



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2000113644 A

(43) Date of publication of application: 21.04.00

(51) Int. CI **G11B 27/00**
G11B 20/12

(21) Application number: 11007946

(71) Applicant: **SONY CORP**

(22) Date of filing: 14.01.99

(72) Inventor: **SHISHIDO YUKIO**

(30) Priority: 06.08.98 JP 10223388

(54) REPRODUCING APPARATUS

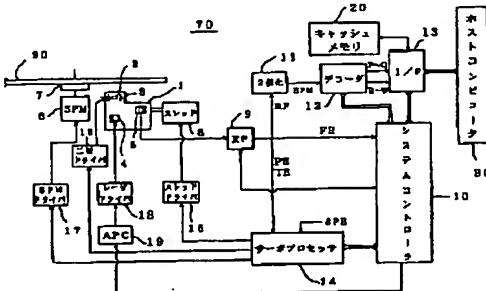
detected start header.

(57) Abstract:

COPYRIGHT: (C)2000,JPO

PROBLEM TO BE SOLVED: To obtain more accurate character information by checking validity of character information corresponding to header information and then improving detection capability through control of error detection of character information when the predetermined data content is detected by verifying various data contents corresponding to the header information indicating character information included in the management information read from a recording medium.

SOLUTION: A reproducing apparatus 70 extracts text data from sub-code data of a disk 90. The sub-code data is sequentially stored in a cache memory 20 as a row data which is not deinterleaved and the detecting process to identify the position of the desired text data from the stored sub-code data is performed. Assuming that the necessary data cannot be obtained only with single read operation, the start header is detected with the detection process from the sub-code data of two text groups stored in the cache memory 20 and the row text data is extracted on the basis of the



(19) 日本国特許庁 (JP)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2000-113644
(P2000-113644A)

(43)公開日 平成12年4月21日(2000.4.21)

(51) Int.Cl.⁷
G 1 1 B 27/00
20/12

識別記号

F I
G 1 1 B 27/00
20/12

テーマコード(参考)
D 5D044
5D110

審査請求 未請求 請求項の数14 O.L (全 32 頁)

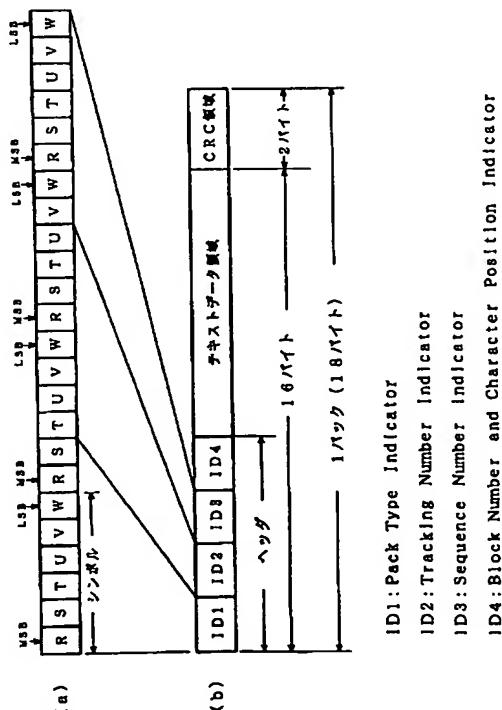
(21)出願番号	特願平11-7946	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22)出願日	平成11年1月14日(1999.1.14)	(72)発明者	宍戸由紀夫 東京都品川区北品川6丁目7番35号 ソニ ー株式会社内
(31)優先権主張番号	特願平10-223388	(74)代理人	100086841 弁理士 脇篤夫 (外1名)
(32)優先日	平成10年8月6日(1998.8.6)	F ターム(参考)	5D044 AB05 AB09 BC03 CC04 DE15 DE18 DE27 DE55 DE83 FG18 FG21 5D110 AA15 BB02 DA10 DB05 DB19 DC06 DE06
(33)優先権主張国	日本(JP)		

(54) 【発明の名称】 再生装置

(57) 【要約】

【課題】 文字情報の開始点と終了点を検出し、文字情報の有効性を確認する。

【解決手段】 ディスクから管理情報を読み込み、この管理情報の中から文字情報の位置を示すスタートヘッダのパターンを少なくとも2個検出する。スタートヘッダはテキストグループ内の先頭のパックのヘッダとされる。これにより、2個のスタートヘッダの間にテキストグループが存在する領域とすることができる（Count1～Count2）。さらに、テキストグループが存在する領域において、所要のパックのヘッダ内容の検証を行ない、テキストデータの有効性を確認する。2個のスタートヘッダが検出されてさえいれば、被検証データとしては各種ヘッダのデータ内容やテキストデータ同士の整合性を検出するようにしても良い。



【特許請求の範囲】

【請求項1】 主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、
前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された、前記管理情報を順次格納することができるようになされている管理情報記憶手段と、
前記管理情報記憶手段に順次格納される前記管理情報に含まれる前記文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、
前記ヘッダ情報検出手段によって検出される第一のヘッダ情報を起点とした順方向における所定の位置のデータ内容を検証する第一の検証手段と、
前記第一のヘッダ情報を統いて検出される第二のヘッダ情報を起点とした逆方向における所定の位置のデータ内容を検証する第二の検証手段と、
前記第一、第二の検証手段の検証結果に基づいて、前記第一、第二のヘッダ情報の間にある前記文字情報が有効であるか否かを判断することができる判断手段と、
を備えたことを特徴とする再生装置。

【請求項2】 前記管理情報記憶手段は、前記管理情報を格納することができる管理情報格納領域と、前記文字情報を格納することができる文字情報格納領域によって形成され、
前記文字情報格納領域には、前記管理情報格納領域に格納された前記管理情報のうち前記文字情報のみが格納されるようにしたことを特徴とする請求項1に記載の再生装置。

【請求項3】 前記記録媒体上の前記管理情報はインターリープされたデータ形態とされているとともに、前記管理情報検出手段はデインターリープされる前の管理情報を検出するようにされていることを特徴とする請求項1に記載の再生装置。

【請求項4】 主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、
前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された前記管理情報を順次格納することができるようになされている管理情報記憶手段と、
前記管理情報記憶手段に順次格納される前記管理情報に含まれる文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、

前記ヘッダ情報検出手段によって検出される第一のヘッダ情報と、該第一のヘッダ情報に統いて検出される第二のヘッダ情報の間の所定の位置にあるデータ内容が、前記文字情報において規定されている所定のデータ内容に

一致しているか否かを検証する検証手段と、
前記検証手段の検証結果に基づいて、前記第一、第二のヘッダ情報の間にある前記文字情報が有効であると判断することができる判断手段と、
を備えたことを特徴とする再生装置。

【請求項5】 前記管理情報記憶手段は、前記管理情報を格納することができる管理情報格納領域と、前記文字情報を格納することができる文字情報格納領域によって形成され、

10 前記文字情報格納領域には、前記管理情報格納領域に格納された前記管理情報のうち前記文字情報のみが格納されるようにしたことを特徴とする請求項4に記載の再生装置。

【請求項6】 前記記録媒体上の前記管理情報はインターリープされたデータ形態とされているとともに、前記管理情報検出手段はデインターリープされる前の管理情報を検出するようにされていることを特徴とする請求項4に記載の再生装置。

【請求項7】 主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、
前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報を順次格納することができるようになされている管理情報記憶手段と、
前記管理情報記憶手段に順次格納される前記管理情報に含まれる前記文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、

30 ヘッダ情報検出手段によって検出される前記ヘッダ情報の間にあるデータを一つの単位として複数単位のデータの内容を比較を行ない、比較したデータの内容が同等であった場合に、当該データが前記文字情報として有効であると判断することができる判断手段と、
を備えたことを特徴とする再生装置。

【請求項8】 前記管理情報記憶手段は、前記管理情報を格納することができる管理情報格納領域と、前記文字情報を格納することができる文字情報格納領域によって形成され、

40 前記文字情報格納領域には、前記管理情報格納領域に格納される前記管理情報のうち前記文字情報のみが格納されるようにしていることを特徴とする請求項7に記載の再生装置。

【請求項9】 前記記録媒体上の前記管理情報はインターリープされたデータ形態とされているとともに、前記管理情報検出手段はデインターリープされる前の管理情報を検出するようにされていることを特徴とする請求項7に記載の再生装置。

【請求項10】 主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出す

ことができる再生手段と、
前記再生手段によって、前記記録媒体から読み出される
管理情報を検出することができる管理情報検出手段と、
前記管理情報検出手段によって検出された前記管理情報
を順次格納することができるようになされている管理情報
記憶手段と、
前記管理情報記憶手段に順次格納される前記管理情報に
含まれる前記文字情報の所要のヘッダ情報を検出するヘ
ッダ情報検出手段と、
前記ヘッダ情報検出手段によって検出される所要のヘッ
ダ情報に基づいて文字情報の検証処理を行なう検証手段
と、
を備え、前記ヘッダ情報検出手段によって所要のヘッダ
情報が2回検出された場合、前記再生手段による前記管
理情報の読み出しを終了するとともに、前記管理情報検
出手段による前記管理情報の検出を終了したうえで、前
記検証手段は前記ヘッダ情報に基づいた前記文字情報の
検証処理を行なうようにしたことを特徴とする再生装
置。

【請求項11】 前記管理情報記憶手段は、前記管理情
報を格納することができる管理情報格納領域と、前記文
字情報を格納することができる文字情報格納領域によ
つて形成され、
前記文字情報格納領域には、前記管理情報格納領域に格
納される前記管理情報のうち前記文字情報のみが格納さ
れるようにされていることを特徴とする請求項10に記
載の再生装置。

【請求項12】 前記記録媒体上の前記管理情報はイン
ターリープされたデータ形態とされているとともに、前
記管理情報検出手段はデインターリープされる前の管理
情報を検出するようにされていることを特徴とする請求
項10に記載の再生装置。

【請求項13】 主データの再生管理情報と文字情報を
含む管理情報が記録された記録媒体から情報を読み出す
ことができる再生手段と、
前記再生手段によって、前記記録媒体から読み出される
管理情報を検出することができる管理情報検出手段と、
前記管理情報検出手段によって検出された前記管理情報
を所定の単位毎に順次格納することができるようにな
れている管理情報記憶手段と、

前記文字情報に対する誤り検出符号に基づいて、前記管
理情報記憶手段に格納される前記管理情報に含まれる前
記文字情報の誤り検出を行う誤り検出手段と、
前記管理情報記憶手段に前記所定の単位とされる前記管
理情報が格納された場合に、格納された前記管理情報に
含まれる前記文字情報の所要のヘッダ情報を検出するヘ
ッダ情報検出手段と、

前記誤り検出手段によって前記管理情報記憶手段に格納
されている前記文字情報に誤りが検出された場合に、前
記ヘッダ情報検出手段によって前記ヘッダ情報が検出さ

れないまま、それまでに前記誤り検出手段によって検出
された前記文字情報の誤り回数が前記管理情報の所定の
単位数以上となったら、前記記録媒体に前記文字情報が
記憶されていないと判別するようにした文字情報有無識
別手段と、

を備えたことを特徴とする再生装置。

【請求項14】 前記文字情報有無識別手段によって、
前記記憶媒体に前記文字情報が記録されていないと判別
した場合は、前記再生手段は前記記録媒体からの前記管
理情報の読み出しを終了するようにしたことを特徴とする
請求項13に記載の再生装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、CD-TEXTに
記憶されている文字情報を再生することができる再生装
置に関するものである。

【0002】

【従来の技術】 例えばオーディオ用のCD(COMPACT DI
SC)のフォーマットに準拠した上で、そのサブコードデ
ータとして所要の文字情報を記憶したCD-TEXTが
知られている。記憶される文字情報としては、例えばデ
ィスクのタイトルやアーティスト名、楽曲名などの情報
とされ、これらの情報を読み出して表示することによつ
て、そのオーディオCDに収録されてる情報の内容など
を文字で得ることが可能である。

【0003】

【発明が解決しようとする課題】 しかし、CD-TEX
Tは、ECCなどのエラー訂正手段を有していないこと
が例えばCD-ROMなどと異なっている。したがつ
て、再生状況によっては、正規のデータが検出されず、
これにより文字情報の始まりと終わりが正しく検出する
ことができないという不具合が生じることがあった。また、
文字情報が前記リードイン領域に記憶されている場
合は、ディスクのアドレスに基づいたリトライ処理を有
効に行なうことができない。即ち、従来では読み出され
たデータにおける文字情報が確かなものか否かを判断す
ることができなかつた。

【0004】

【課題を解決するための手段】 本発明はこのような問題
点を解決するために、主データの再生管理情報と文字情
報を含む管理情報が記録された記録媒体から情報を読み
出すことができる再生手段と、前記再生手段によって、
前記記録媒体から読み出される管理情報を検出する
ことができる管理情報検出手段と、前記管理情報検出手
段によって検出された、前記管理情報を順次格納する
ことができるようになされている管理情報記憶手段と、前記管
理情報記憶手段に順次格納される前記管理情報に含まれる
前記文字情報の所要のヘッダ情報を検出するヘッダ情報
検出手段と、前記ヘッダ情報検出手段によって検出され
る第一のヘッダ情報を起点とした順方向における所定の

位置のデータ内容を検証する第一の検証手段と、前記第一のヘッダ情報に統いて検出される第二のヘッダ情報を起点とした逆方向における所定の位置のデータ内容を検証する第二の検証手段と、前記第一、第二の検証手段の検証結果に基づいて、前記第一、第二のヘッダ情報の間にある前記文字情報が有効であるか否かを判断することができる判断手段を備えて再生装置を構成する。

【0005】また、主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された前記管理情報を順次格納することができるようになっている管理情報記憶手段と、前記管理情報記憶手段に順次格納される前記管理情報に含まれる文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、前記ヘッダ情報検出手段によって検出される第一のヘッダ情報と、該第一のヘッダ情報に統いて検出される第二のヘッダ情報の間に所定の位置にあるデータ内容が、前記文字情報において規定されている所定のデータ内容に一致しているか否かを検証する検証手段と、前記検証手段の検証結果に基づいて、前記第一、第二のヘッダ情報の間にある前記文字情報が有効であると判断することができる判断手段を備えて再生装置を構成する。

【0006】また、主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された前記管理情報を順次格納することができるようになっている管理情報記憶手段と、前記管理情報記憶手段に順次格納される前記管理情報に含まれる前記文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、ヘッダ情報検出手段によって検出される前記ヘッダ情報の間にデータを一つの単位として複数単位のデータの内容を比較を行ない、比較したデータの内容が同等であった場合に、当該データが前記文字情報として有効であると判断することができる判断手段を備えて再生装置を構成する。

【0007】さらに、主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された前記管理情報を順次格納することができるようになっている管理情報記憶手段と、前記管理情報記憶手段に順次格納される前記管理情報に含まれる前記文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、前記ヘッダ情報検出手段によって検出される所要のヘッダ情報に基づいて文字情報の検証処理を行な

う検証手段を備え、前記ヘッダ情報検出手段によって所要のヘッダ情報が2回検出された場合、前記再生手段による前記管理情報の読み出しを終了するとともに、前記管理情報検出手段による前記管理情報の検出を終了したうえで、前記検証手段は前記ヘッダ情報に基づいた前記文字情報の検証処理を行なうように再生装置を構成する。

【0008】また、主データの再生管理情報と文字情報を含む管理情報が記録された記録媒体から情報を読み出すことができる再生手段と、前記再生手段によって、前記記録媒体から読み出される管理情報を検出することができる管理情報検出手段と、前記管理情報検出手段によって検出された前記管理情報を所定の単位毎に順次格納することができるようになっている管理情報記憶手段と、前記文字情報に対する誤り検出符号に基づいて、前記管理情報記憶手段に格納される前記管理情報に含まれる前記文字情報の誤り検出を行う誤り検出手段と、前記管理情報記憶手段に前記所定の単位とされる前記管理情報が格納された場合に、格納された前記管理情報に含まれる前記文字情報の所要のヘッダ情報を検出するヘッダ情報検出手段と、前記誤り検出手段によって前記管理情報記憶手段に格納されている前記文字情報に誤りが検出された場合に、前記ヘッダ情報検出手段によって前記ヘッダ情報が検出されないまま、それまでに前記誤り検出手段によって検出された前記文字情報の誤り回数が前記管理情報の所定の単位数以上となったら、前記記録媒体に前記文字情報が記憶されていないと判別するようにした文字情報有無識別手段を備えて再生装置を構成する。

【0009】本発明によれば、記録媒体から読み込んだ管理情報から、この管理情報に含まれる文字情報を示すヘッダ情報に対応した各種データ内容を検証し、所定のデータ内容が検出され場合に、当該ヘッダ情報に対応した文字情報の有効性を確認することができるようになっている。これにより、文字情報の誤検出を抑制し検出能力を向上させることができ、より正確な文字情報を得ることができるようになる。

【0010】また、所定数のヘッダ情報が2回検出された時点での管理情報の検出を終了したうえで文字情報の検証処理を行なうようにすることにより、それ以降行なう検証処理の処理負担を軽減することができるようになる。

【0011】さらに、前記文字情報に対する誤り検出符号に基づいて、当該文字情報に誤りが検出されたときに、ヘッダ情報が検出されないまま、それまでに前記文字情報の誤りが所定回数以上検出されていた場合には、記録媒体に文字情報が記録されていないと判断するようになっているので、必要以上の再生動作を抑制したうえで、記録媒体の種別を識別することができるようになる。

【0012】

【発明の実施の形態】以下、本発明の実施の形態として、CD-TEXTとしてのディスクに対応して、オーディオデータ及びサブコードとしての文字情報を再生することができる再生装置を例に挙げて次に示す順序で説明する。なお、サブコードとして記録される文字情報を「テキストデータ」ともいうこととする。

<1. 再生装置の構成>

<2. TOC及びサブコード>

<3. テキストデータ>

<4. 再生処理>

<5. パファリング処理>

<6. 8ビット／6ビット変換>

<7. テキストデータ有効性検証処理>

<8. データ例>

<9. 変形例>

【0013】1. 再生装置の構成

本実施の形態の再生装置に装填される光ディスクは、例えばCD-TEXTなどのCD方式のディスクや、DVDなどが考えられる。もちろん他の種類の光ディスクに対応する再生装置でも本発明は適用できるものである。

【0014】図1は本例の再生装置70の要部のブロック図である。ディスク90は、ターンテーブル7に積載され、再生動作時においてスピンドルモータ6によって一定線速度(CLV)もしくは一定角速度(CAV)で回転駆動される。そしてピックアップ1によってディスク90にエンボスピット形態や相変化ピット形態などで記録されているデータの読み出しが行なわれることになる。なお本例ではCLV方式として説明を続ける。

【0015】ピックアップ1内には、レーザ光源となるレーザダイオード4や、反射光を検出するためのフォトディテクタ5、レーザ光の出力端となる対物レンズ2、レーザ光を対物レンズ2を介してディスク記録面に照射し、またその反射光をフォトディテクタ5に導く光学系が形成される。対物レンズ2は二軸機構3によってトラッキング方向及びフォーカス方向に移動可能に保持されている。またピックアップ1全体はスレッド機構8によりディスク半径方向に移動可能とされている。

【0016】ディスク90からの反射光情報はフォトディテクタ5によって検出され、受光光量に応じた電気信号とされてRFアンプ9に供給される。RFアンプ9には、フォトディテクタ5としての複数の受光素子からの出力電流に対応して電流電圧変換回路、マトリクス演算／增幅回路等を備え、マトリクス演算処理により必要な信号を生成する。例えば再生データであるRF信号、サーボ制御のためのフォーカスエラー信号FE、トラッキングエラー信号TEなどを生成する。RFアンプ9から出力される再生RF信号は2値化回路11へ、フォーカスエラー信号FE、トラッキングエラー信号TEはサーボプロセッサ14へ供給される。

【0017】RFアンプ9で得られた再生RF信号は2

値化回路11で2値化されることでいわゆるEFM信号(8-14変調信号；CDの場合)もしくはEFM+信号(8-16変調信号；DVDの場合)とされ、デコーダ12に供給される。デコーダ12ではEFM復調、エラー訂正処理等を行ない、また必要に応じてCD-TEXTデコード、CD-ROMデコード、またはMPEGデコードなどを行なってディスク90から読み取られた情報の再生を行なう。

【0018】なおデコーダ12は、EFM復調したデータをデータバッファとしてのキャッシングメモリ20に蓄積していく、このキャッシングメモリ20上でエラー訂正処理等を行う。そしてエラー訂正され適正な再生データとされた状態で、キャッシングメモリ20へのパファリングが完了される。また、本例ではCD-TEXTデータを検証する際のローデータ(RAW_DATA・・・ディスク90から読み出されインターりーブされていないデータ)のバッファリングエリアとしても用いられる。なお、ローデータのバッファリング処理については後述する。再生装置70からの再生出力としては、キャッシングメモリ20でパファリングされているデータが読み出されて転送出力されることになる。

【0019】インターフェース部13は、外部のホストコンピュータ80と接続され、ホストコンピュータ80との間で再生データやリードコマンド等の通信を行う。即ちキャッシングメモリ20に格納された再生データは、インターフェース部13を介してホストコンピュータ80に転送出力される。またホストコンピュータ80からのリードコマンドその他の信号はインターフェース部13を介してシステムコントローラ10に供給される。

【0020】サーボプロセッサ14は、RFアンプ9からのフォーカスエラー信号FE、トラッキングエラー信号TEや、デコーダ12もしくはシステムコントローラ10からのスピンドルエラー信号SPE等から、フォーカス、トラッキング、スレッド、スピンドルの各種サーボドライブ信号を生成しサーボ動作を実行させる。即ちフォーカスエラー信号FE、トラッキングエラー信号TEに応じてフォーカスドライブ信号、トラッキングドライブ信号を生成し、二軸ドライバ16に供給する。二軸ドライバ16はピックアップ1における二軸機構3のフォーカスコイル、トラッキングコイルを駆動することになる。これによってピックアップ1、RFアンプ9、サーボプロセッサ14、二軸ドライバ16、二軸機構3によるトラッキングサーボループ及びフォーカスサーボループが形成される。

【0021】またサーボプロセッサ14はスピンドルモータドライバ17に対して、スピンドルエラー信号SPEに応じて生成したスピンドルドライブ信号を供給する。スピンドルモータドライバ17はスピンドルドライブ信号に応じて例えば3相駆動信号をスピンドルモータ6に印加し、スピンドルモータ6のCLV回転を実行さ

せる。またサーボプロセッサ14はシステムコントローラ10からのスピンドルキック／ブレーキ制御信号に応じてスピンドルドライブ信号を発生させ、スピンドルモータドライバ17によるスピンドルモータ6の起動、停止、加速、減速などの動作も実行させる。

【0022】なお、スピンドルモータ6のCLV回転としての線速度については、システムコントローラ10が各種速度に設定できる。例えばデコーダ12は、デコード処理に用いるためにE FM信号に同期した再生クロックを生成するが、この再生クロックから現在の回転速度情報を得ることができる。システムコントローラ10もしくはデコーダ12は、このような現在の回転速度情報と、基準速度情報を比較することで、CLVサーボのためのスピンドルエラー信号SPEを生成する。従って、システムコントローラ11は、基準速度情報としての値を切り換えれば、CLV回転としての線速度を変化させることができる。例えばある通常の線速度を基準として4倍速、8倍速などの線速度を実現できる。これによりデータ転送レートの高速化が可能となる。なお、もちろんCAV方式であっても回転速度の切換は可能である。

【0023】サーボプロセッサ14は、例えばトラッキングエラー信号TEの低域成分として得られるスレッドエラー信号や、システムコントローラ10からのアクセス実行制御などに基づいてスレッドドライブ信号を生成し、スレッドドライバ15に供給する。スレッドドライバ15はスレッドドライブ信号に応じてスレッド機構8を駆動する。スレッド機構8には図示しないが、ピックアップ1を保持するメインシャフト、スレッドモータ、伝達ギア等による機構を有し、スレッドドライバ15がスレッドドライブ信号に応じてスレッドモータ8を駆動することで、ピックアップ1の所要のスライド移動が行なわれる。

【0024】ピックアップ1におけるレーザダイオード4はレーザドライバ18によってレーザ発光駆動される。システムコントローラ10はディスク90に対する再生動作を実行させる際に、レーザパワーの制御値をオートパワーコントロール回路19にセットし、オートパワーコントロール回路19はセットされたレーザパワーの値に応じてレーザ出力が行われるようにレーザドライバ18を制御する。

【0025】以上のようなサーボ及びデコード、エンコードなどの各種動作はマイクロコンピュータによって形成されたシステムコントローラ10により制御される。そしてシステムコントローラ10は、ホストコンピュータ80からのコマンドに応じて各種処理を実行する。例えばホストコンピュータ80から、ディスク90に記録されている或るデータの転送を求めるリードコマンドが供給された場合は、まず指示されたアドレスを目的としてシーク動作制御を行う。即ちサーボプロセッサ14に指令を出し、シークコマンドにより指定されたアドレス

をターゲットとするピックアップ1のアクセス動作を実行させる。その後、その指示されたデータ区間のデータをホストコンピュータ80に転送するために必要な動作制御を行う。即ちディスク90からのデータ読出／デコード／バッファリング等を行って、要求されたデータを転送する。

【0026】ホストコンピュータ80からのリードコマンド、即ち転送要求としては、要求するデータ区間の最初のアドレスとなる要求スタートアドレスと、その最初のアドレスからの区間長として要求データ長（データレンジス）となる。例えば要求スタートアドレス=N、要求データ長=3という転送要求は、LBA「N」～LBA「N+2」の3セクターのデータ転送要求を意味する。LBAとは論理ブロックアドレス（LOGICAL BLOCK ADDRESS）であり、ディスク90のデータセクターに対して与えられているアドレスである。

【0027】2. TOC及びサブコード

次に、ディスク90においてリードインエリアに記録されるTOC、及びサブコードについて説明する。ディスク90において記録されるデータの最小単位は1フレームとなる。98フレームで1ブロック（1サブコードディングフレーム）が構成される。

【0028】1フレームの構造は図2のようになる。1フレームは588ビットで構成され、先頭24ビットが同期データと、これに続く3ビットによるマージンビットが設定され、続いて14ビットがサブコードデータエリアとされる。そして、その後に12シンボルのメインデータ及び4シンボルのパリティデータが配される。

【0029】この構成のフレームが98フレームで1ブロックが構成され、98個のフレームから取り出されたサブコードデータが集められて図3(a)のような1ブロックのサブコードデータが形成される。98フレームの先頭の第1、第2のフレーム（フレーム98n+1、フレーム98n+2）からのサブコードデータは同期パターンとされている。そして、第3フレームから第98フレーム（フレーム98n+3～フレーム98n+98）まで、各96ビットのチャンネルデータ、即ちP, Q, R, S, T, U, V, Wのサブコードデータが形成される。

【0030】このうち、アクセス等、再生に関わる各種制御には再生管理情報とされるPチャンネルとQチャンネルが用いられる。但し、Pチャンネルはトラックとトラックの間のポーズ部分を示しているのみで、より細かい制御はQチャンネル（Q1～Q96）によって行なわれる。96ビットのQチャンネルデータは図3(b)のように構成される。Rチャンネル～Wチャンネルのデータは、テキストデータ群を形成するために設けられるが、これについては後述する。

【0031】まずQ1～Q4の4ビットはコントロール

50 データとされ、オーディオのチャンネル数、エンファシ

ス、CD-ROMの識別などに用いられる。即ち、4ビットのコントロールデータは次のように定義される。

【0***】 …… 2チャンネルオーディオ
 【1***】 …… 4チャンネルオーディオ
 【*0**】 …… CD-DA (CDデジタルオーディオ; CD-TEXTを含む)
 【*1**】 …… CD-ROM
 【**0*】 …… デジタルコピー不可
 【**1*】 …… デジタルコピー可
 【***0】 …… プリエンファシスなし
 【***1】 …… プリエンファシスあり
 【0032】 次にQ5～Q8の4ビットはアドレスとされ、これはサブQデータのコントロールビットとされている。このアドレス4ビットが「0001」である場合は、続くQ9～Q80のサブQデータはオーディオQデータであることを示し、また「0100」である場合は、続くQ9～Q80のサブQデータがビデオQデータであることを示している。そしてQ9～Q80で72ビットのサブQデータとされ、残りのQ81～Q96はCRCとされる。
 【0033】 リードインエリアにおいては、そこに記録されているサブQデータが即ちTOC情報となる。つまりリードインエリアから読み込まれたQチャンネルデータにおけるQ9～Q80の72ビットのサブQデータは、図4(a)のような情報を有するものである。サブQデータは各8ビットのデータを有している。
 【0034】 まずトラックナンバが記録される。リードインエリアではトラックナンバは「00」に固定される。続いてPOINT(ポイント)が記され、さらにトラック内の経過時間としてMIN(分)、SEC(秒)、FRAME(フレーム番号)が示される。さらに、PMIN, PSEC, PFRAMEが記録されるが、このPMIN, PSEC, PFRAMEは、POINTの値によって、次に述べるように意味が決定されている。
 【0035】 POINTの値が「01h」～「99h」(hは16進表現であることを示す)のときは、その値はトラックナンバを意味し、この場合PMIN, PSEC, PFRAMEにおいては、そのトラックナンバのトラックのスタートポイント(絶対時間アドレス)が分(PMIN), 秒(PSEC), フレーム番号(PFRAME)として記録されている。
 【0036】 POINTの値が「A0h」のときは、PMINに最初のトラックのトラックナンバが記録される。また、PSECの値によってCD-DA, CD-I, CD-ROM(XA仕様)が区別される。POINTの値が「A1h」のときは、PMINに最後のトラックのトラックナンバが記録される。POINTの値が「A2h」のときは、PMIN, PSEC, PFRAMEにリードアウトエリアのスタートポイントが絶対時間

アドレスとして示される。

【0037】 例えば6トラックが記録されたディスクの場合、このようなサブQデータによるTOCとしては図5のようにデータが記録されていることになる。図5に示すようにトラックナンバTNOは全て「00h」である。BLOCK NO. とは上記のように98フレームによるBLOCKデータとして読み込まれた1単位のサブQデータのナンバを示している。各TOCデータはそれぞれ3BLOCKにわたって同一内容が書かれている。図示するようにPOINTが「01h」～「06h」の場合、PMIN, PSEC, PFRAMEとしてトラック#1～トラック#6のスタートポイントが示されている。

【0038】 そしてPOINTが「A0h」の場合、PMINに最初のトラックナンバとして「01」が示される。またPSECの値によってディスクが識別され、このディスクがCD-DAの場合は、図示するようにPSEC = 「00h」とされる。なお、CD-ROM(XA仕様)の場合は、PSEC = 「20h」、CD-Iの場合には「10h」となる。

【0039】 そしてPOINTの値として「A1h」の位置にPMINに最後のトラックのトラックナンバが記録され、さらにPOINTの値として「A2h」の位置には、PMIN, PSEC, PFRAMEにそれぞれリードアウトエリアのスタートポイントが示される。BLOCK n+27以降は、BLOCK n～n+26の内容が再び繰り返して記録されている。

【0040】 ディスク90上で実際に音楽等のデータが記録されるトラック#1～#n、及びリードアウトエリアにおいては、そこに記録されているサブQデータは図4(b)に示されている情報を有する。まずトラックナンバが記録される。即ち各トラック#1～#nでは「01h」～「99h」のいづれかの値となる。またリードアウトエリアではトラックナンバは「AAh」とされる。続いてインデックスとして各トラックをさらに細分化することができる情報が記録される。

【0041】 そして、トラック内の経過時間としてMIN(分)、SEC(秒)、FRAME(フレーム番号)が示される。さらに、AMIN, ASEC, AFRAMEとして、絶対時間アドレスが分(AMIN), 秒(ASEC), フレーム番号(AFRAME)として記録されている。

【0042】 このようにTOC及びサブコードが形成されているわけであるが、ディスク上のアドレス、即ちAMIN, ASEC, AFRAMEは、98フレーム単位で記録されることが理解される。この98フレーム(1BLOCK)は1サブコーディングフレームと呼ばれ、音声データとしての1秒間には75サブコーディングフレームが含まれることになる。つまり、アドレスとしての「AFRAME」がとりうる値は「0」～「74」となる。なお、後述するフレームチェック処理でデータの連

統性がチェックされるのは、このサブコーディングフレーム単位となる。

【0043】3. テキストデータ

以降、図2及び図3に示す構造のサブコードに含まれるテキストデータについて説明を行うこととし、まず、図6によりテキストデータの包括的な構造について説明する。サブコード内に含まれるテキストデータのみを抽出して構造的に見た場合、テキストデータは図6に示すようなものとなる。テキストデータとしての最も大きなデータ単位は、図6(a)に示す「テキストグループ」とされる。図6(a)において複数のテキストグループが示されているが、各テキストグループのデータは同一内容とされており、従って、サブコード内においては、同一データ内容の所定数の複数のテキストグループが繰り返し記録されていることになる。

【0044】1つのテキストグループは、例えば最大2048パック(パックの定義については後述する)により形成するものとされるが、1テキストグループあたりのデータ読出しに要する時間等を考慮して、1テキストグループを512パック以内により形成することが推奨されている。この際、1テキストグループあたりのデータ総量としては6500文字程度となる。また1つのテキストグループは、図6(b)に示すようにブロック#0～ブロック#nにより形成され、例えば最大8ブロック(0≤n≤7となる)であると規定されている。各ブロックは、同一の内容の情報をそれぞれ異なる言語により表記するためのテキストデータが格納されているものとされる。例えば、ブロック#0には、当該ディスクに対応する各種情報を英語により表記するためのテキストデータが格納されており、ブロック#1には、ブロック#0と内容的には同一の情報を日本語により表記するためのテキストデータが格納されているものとされる。

【0045】この場合、1テキストグループは最大8ブロックにより形成可能であることから、テキストデータのフォーマットとしては最大8言語に対応することが可能とされる。1つのブロックは、図6(c)に示すようにパック#0～パック#nのデータ単位により形成される。ここでは、1ブロックは最大256パックで形成され情報量としては例えば36.864KByteとされている。なお、パック内のデータ構造等については、次の図7、図8及び図9により説明する。

【0046】図7(a)は、図3に示した1サブコーディングフレームをデータ領域別に示すものであり、前述のように1サブコーディングフレームは98フレームにより形成される。98フレームの先頭の第1、第2フレーム(フレーム98n+1、フレーム98n+2)は、図3で説明したように同期パターンS0、S1の領域とされる。また、第3フレームから第98フレーム(フレーム98n+3～フレーム98n+98)におけるPチャンネルはサブコードPのデータ領域とされ、Qチャン

ネルはサブコードQのデータ領域とされて、前述したようにアクセス等の管理のためのデータが格納される。

【0047】そして、第3フレームから第98フレームにおけるRチャンネル～Wチャンネルの領域は図のようにパック0～パック4とされる。各パックのデータサイズは固定長とされて、図7(b)に示すようにシンボル0～23の24シンボルにより形成される。1シンボルは図7(c)に示すように、1フレームにおけるR、S、T、U、V、Wのチャンネルデータよりなる6ビットのデータ単位であり、この場合にはRチャンネルデータがMSB、WチャンネルがLSBとして定義される。

【0048】図8は、上記図7(a)に示す構造の1サブコーディングフレームから、4つのパック(パック0～パック4)によるデータ構造を抜き出して示している。1パックは、図7で説明したように、24のシンボル(6ビット)により形成されることから、

$$6\text{ビット} \times 24 / 8 = 18\text{バイト}$$

で示されるように18バイトのデータサイズを有する。そして、図のように1パックは、先頭のID領域と続く

テキストデータ領域により16バイトを占有し、残りの2バイトはCRC領域となる。また、前述のように1サブコーディングフレームにおいては4つのパックが設けられるが、これら4つのパックの集合により形成されるデータ単位はパケットとして定義されている。1パックは24シンボルにより形成されることから、1パケットは、

$$24\text{ (シンボル)} \times 4\text{ (パック)} = 96\text{ (シンボル)}$$

で示されるように96のシンボルにより形成されるものとみることができる。

【0049】図9及び図10は、図8で示した1パック分のデータをシリアルに表現したものである。図9(a)から分かるように、本実施の形態におけるテキストデータのフォーマットにおいては、6ビットよりなるシンボルをシリアルに配列させたうえで、このデータ列を8ビット(1バイト)ごとに区切るようにしてデータを扱うように規定されている。

【0050】本実施の形態のテキストデータのフォーマットでは、図9(b)及び図10に示すように、パックの先頭からはヘッダ領域としてID1、ID2、ID

3、ID4の4つのIDデータが設けられる。なお、ID1はPack_Type_Indicator、ID2はTrack_Number_Indicator、ID3はSequence_Number_Indicator、ID4はBlock_Number_and_Character_Position_Indicatorとされているが、これらの内容については後述する。

【0051】本実施の形態のフォーマットとして8ビット(1バイト)ごとに区切ってデータを扱うことにより、これら各IDはそれぞれ8ビット(1バイト)のデ

ータ単位とされることになる。このため、図9 (b) に示すように、ヘッダ領域 (ID1～ID4) 以降の残りの12バイトがテキストデータ領域として確保され、残りの2バイトがCRC領域となる。そして、上記12バイトのテキストデータ領域も、図10に示すパックの構造図に示すように、8ビットごとのtext1～text12のデータ単位により扱われるものとされる。なお、以降の説明では、単に「ヘッダ」という場合はパックの先頭にあるヘッダ領域 (ID1～ID4) を示し、テキストグループ内における先頭のパックのヘッダについては「スタートヘッダ」とすることとする。

【0052】ここで、本実施の形態のテキストデータのフォーマットにおいては、図11に示すようにテキストデータ対応以外のCDのフォーマットに準じて、パックの先頭のID1の上位3ビットをモード(MODE)として扱い、続く3ビットをアイテム(ITEM)として扱うことができるようしている。そして、上位3ビットのモードとしては、モード4を設定してこの3ビットに対して値「100」を設定するようしている。このモード4はCD-TEXTフォーマットが提案される前は未定義とされている。こうすることで、例えばテキストデータに対応していない再生装置等にテキストデータが格納されたCDを装填しても、モードとして認識不可能とされることでその動作を停止するだけであって、誤動作することはないようにされる。また、モード4としてはテキストデータがリードインエリアに記憶されているモードであるが、この他にも、プログラムエリア(主データが格納されているエリア)に対してテキストデータが記憶されているモード2が知られている。本例ではモード4を例に挙げて説明するが、本発明はモード2を適用している場合にも対応することが可能とされる。

【0053】なお、未定義のモードとしてはモード5及びモード6が存在するので、モード4の代わりにこれらのモードに対応する値を設定することも可能である。また、参考までに、使用済みのモードとしては、CD-Gに対応するモード1、CD-MIDIに対応するモード3等が存在する。なお、ここではアイテム(ITEM)としての値は特に設定されず、後述するように、ID1により定義する識別内容に応じて下位3ビット以降の値は適宜異なる(実際には下位4ビットのみが変更される)ものとなる。

【0054】次に、図12及び図13を参照して、本実施の形態が対応するテキストデータフォーマットにおけるヘッダ領域(ID1、ID2、ID3、及びID4)の各定義内容について説明する。図12(a)～(d)はそれぞれID1～ID4を示し、図13はID1の定義内容を示している。図12(a)に示すID1(8ビット)は、当該パックのテキストデータ領域におけるtext1以降に格納される文字列の内容の種類を識別するためのデータが設定されるものであり、「80h」～

「8Fh」の値をとるものとされる。ここで、ID1について設定される値として上位4ビットが16進法によりすべて「8」が設定されているのは、図11で説明したように、ID1の上位3ビットをモード(MODE)として扱った場合に「100」の値が得られるようにして、モード4として識別できるようにするためである。

【0055】ID1に設定される値「80h」～「8Fh」に対応する定義内容は、図13に示すように規定されている。この図によると、ID1が「80h」の場合には、text1以降に格納される文字列の内容が、アルバムタイトル(Tittle Of Album Name・・・ID2が「00h」の場合)又はトラックに記録された楽曲等の曲名(Track Title・・・ID2が「01h～63h」の場合)であることを示すことになる。また、ID1が「81h」の場合には演奏者、指揮者、又はオーケストラ名であることを示し、

【0056】「82h」の場合には作詞者名、「83h」の場合には作曲者名、「84h」の場合には編曲者名であることを示す。ID1が「85h」の場合には、当該CDを供給する者(例えばレコード会社等)や演奏者などからのメッセージであることが示される。

【0057】また、ID1が「86h」の場合には、例えばカタログナンバーやレコード会社の名前等により決められるディスクIDであることが示され、「87h」の場合にはジャンルを示すテキストデータであることが示され、「88h」の場合にはTOCデータであることが示される。このTOCデータは、例えばQチャンネルのサブコードデータに準ずる内容を示すものとなる。また、「89h」の場合には、2nd TOCであることが示される。ID1として「8Ah」「8Bh」「8Ch」は予約 RESERVEDとされている。

【0058】図12(b)に示すID2は、当該パックのテキストデータ領域におけるtext1以降に格納される文字列がどのトラックに対応するのかをトラックナンバにより示すものとされ、ID2を形成する8ビットにより、「00h」～「63h」(10進法では0～99)の値をとるものとされる。但し、トラックナンバは「1」からインクリメントされるようにして付されていくものであるため、トラックナンバとしては「01h」～「63h」(10進法では1～99)の値をとることになる。値「00h」は、ディスク全体を代表することを意味するものとされる。ID2のMSBは、拡張用フラグとされているがここでは常に「0」を設定するもの

とされ、「1」が設定されると拡張用のフラグが立てられたことになる。

【0059】図12(c)に示すID3は、当該パックが属するブロック内において、当該パックが何番目のパックであるのかを示すパックのブロック内連番を示すものとされ、ID3を形成する8ビットにより、「00h」～「FFh」(10進法では0～255)の値をとるものとされる。

【0060】図12(d)に示すID4は、現パックのブロック番号(文字コードの識別情報を含む)と、1まとまりの文字列の文字位置を示すものとされる。MSBは当該パックにおけるテキストデータが1バイトコードか2バイトコードであるかを示す2バイトコードフラグの領域とされ、値として「1」の場合には2バイトコードであることが示され、「0」の場合には1バイトコードであることが示される。MSBに続く上位第2ビットから第4ビットまでの3ビットは、当該パックを含むブロック(図6(b)参照)のブロック番号が示され、2進法で「000」～「111」(10進法では0～7)の値を取るものとされる。図6により説明したようにブロックは最大8つ設けられて、ブロックナンバとしてはブロックナンバ0～7の値を取り得ることになるが、上記3ビットにより取り得る値はこれに対応したものとなる。

【0061】ところで、現状として、少なくともブロック#0においては、テキストデータとしてASCIIコードを含む8859-1コードのみを使用することが規定されている。つまり、ブロック#0においては、一般的には言語として英語による表記を行うためのテキストデータが格納されることになる。なお、以降の説明においては、便宜上、ブロック0は言語として英語に対応し、文字コードとしてはASCIIコードを用いるものとする。ASCIIコード(及び8859-1コード)は1バイトコードであることから、ブロック#0に含まれる全パック内のID4においては、上位4ビットは「0000」となる。

【0062】ID4の下位4ビットは、現パックにおける文字位置の情報が格納される。つまり、現パックのテキストデータ領域における最初のtext1に格納されている文字データが、1まとまりの文字列における何番目の文字であるのかを示すものとされ、図13(d)に示すように、2進法で「0000」～「1111」の値をとる。なお、16番目以上の文字である場合にはすべて「1111」となる。また、ここでいう「1まとまりの文字列」とは、例えば、トラックの曲名データであれば、この1トラック分の曲名を形成する一連の文字列を意味するものである。

【0063】図14には、トラックごとの曲名を示すテキストデータをテキストデータ領域に格納する場合の1パックの構造例が示されている。この場合、図12

(a) 及び図13により説明したように、ID1は「80h」とされ、ID2には、当該パックのテキストデータにより表記するタイトルのトラックに対応するトラックナンバが「01h」～「63h」(トラック1～99)により適宜示される。ID3には、ブロック内における当該パックのブロック内連番が「00h」～「FFh」により示される。ID4は当該パックを含む現ブロックのブロック番号(図6(b)参照)が第2ビット～第4ビットの3ビットにより示され、現ブロックが対応する文字コードが2バイトコードか1バイトコードであるかがMSBにより示される。例えば、現パックのテキストデータがASCIIコードに対応するものであれば、前述のようにID4の上位4ビットは「0000」となる。

【0064】また、ID4の下位4ビットは、text1に格納されている文字データが、1まとまりの文字列における何番目の文字であるのかを示す値が適宜示されることになる。トラックごとの曲名を示すテキストデータの場合、上記「1まとまりの文字列」とは、トラックごとの曲名に対応する文字列がこれに相当するものとされ、例えば、あるトラックの曲名が「THIS_IS_APEN」であるとして、このTHIS_IS_APENの文字列のうち、2番目の「H」の文字データが当該パックのtext1に格納されているとすれば、当該パックのID4における下位4ビットは「0001(1h)」とされることになる。

【0065】この場合、「THIS_IS_APEN」のうち最初の文字データ「T」は、当該パックの直前のパックのテキストデータ領域に格納されていることになる。つまりテキストデータは、1まとまりの文字列データが連続するパックをまたがるようにしてテキストデータ領域に格納することが可能とされるフォーマットとされている。そして、テキストデータ領域である各8ビットのtext1～text12には、本実施の形態のテキストデータフォーマットに則った規則に従って、各トラックの曲名を示す文字コードのデータが格納されることになる。

【0066】4. 再生処理

本実施の形態におけるテキストデータの再生処理の概要を説明する。まず、ここでは全体の流れを説明し、詳細な説明については順次行なっていくこととする。

【0067】上記したように、「モード4」においてはテキストデータはディスク90のサブコードデータの一部に記録されている。したがって、テキストデータを扱う場合は、まずサブコードデータ(P～W)から不要なデータ(P、Q)を排除してテキストデータ(R～W)の抽出を行なうことになる。この場合サブコードデータはデインターリープを行なわないローデータとしてキャッシュメモリ20に順次格納していく。ローデータとしては96バイト(96bit×8(P～W))、即ち1

サブコードフレームに相当するデータ量が1回のバファリング単位とされる。そして、キャッシュメモリ20に格納されたサブコードデータの中から、所望するテキストデータの位置を識別する検出処理を行なう。この検出処理としてはテキストグループのスタートヘッダを検出するための処理とされる。以降、この処理をバファリング処理という。

【0068】バッファリング処理としては、サブコードデータにおいて図6(a)に示したテキストグループの先頭とされる位置(先頭パックのヘッダに相当する)を起点とし読み出しを行なうこと望ましいが、実際にはこれは困難な処理とされる。そこで、1回の読み込みではバファリング処理を行なうに際して必要なデータが得られない場合を想定して、キャッシュメモリ20には少なくとも2テキストグループ分のデータ量に相当するサブコードデータを格納することができるようにして、この格納された2テキストグループ分のサブコードデータの中からスタートヘッダを検出するようにしている。

【0069】バッファリング処理を行なう場合、キャッシュメモリ20は例えば図15に示されている形態でマッピングされた状態で使用する。ローサブコードエリア(RAW_SUB_CODE_AREA)は、例えば2テキストグループ分のデータ量に相当するローサブコードデータ(RAW_SUB_CODE_DATA(P~W))を格納することができる領域とされる。そして、後述するようにローサブコードエリアに格納されたサブコードデータに対してバッファリング処理を行ない、このバファリング処理によって検出された、スタートヘッダに基づいてローテキストデータ(RAW_TEXT_DATA(R~W))の抽出(8ビット(P~W)から6ビット(R~W)への変換処理・・・以下、8ビット/6ビット変換処理という)を行なう。ここで、抽出されたローテキストデータは、図15に示すローテキストエリア(RAW_TEXT_AREA)に格納される段階で、データ内容の有効性が検証される。有効性が確認されたローテキストデータは、ローテキストエリアから読み出されてホストコンピュータ80に対して転送されるようにされる。即ち、ホストコンピュータ80に対しては、有効性が確認されたローテキストデータのみを転送することができるようになる。

【0070】ところで、図示している各エリアの容量を示す数値は一例である。前記したように、テキストグループの1ブロックの情報量は上述したように例えば36.864Kバイトとされる。したがって、ローテキストエリアに必要な容量としては少なくとも、 $36.864 \times 2 = 73.728$ Kバイト以上であれば良い。また、ローサブコードエリアは、 $36.864 \times 8 / 6 = 49.152$ Kバイトが1フレーム分のデータ量に相当するので、2フレームに相当する容量としては $49.152 \times 2 = 98.304$ Kbyteとなる。したがつ

て、ローサブコードエリアに必要な容量は、少なくとも98.304Kbyte以上であれば良い。

【0071】5. バファリング処理

図16、図17は、バッファリング処理の流れを模式的に説明するフローチャートを示す図である。なお、図16に示されているフローチャートの結合子(1)は、図17に示されているフローチャートの結合子(1)に対応したものである。即ち、バファリング処理としては図16乃至図17に示されている一連の処理行程によって10 実現される処理とされる。したがって、各図には連続したステップ番号を付している。

【0072】バッファリング処理の開始を指示するコマンドはホストコンピュータ80から供給される(S001)。ここで、再生装置が受信するコマンドとしては、例えば「A0コマンド」、「Packetコマンド」などのリードTOCコマンドを拡張してCD-TEXTに格納されたテキストデータを読み込む処理を開始するためのコマンドとされる。このようなコマンドを受信すると、「OPコードチェック」、「CDBチェック」、20 「メディアチェック」などの各種チェック処理が行なわれる(S002)。

【0073】「OPコードチェック」によって、受信したコマンドのOPコードが当該装置に対応しているものかの可否判断が行なわれ、「CDBチェック」によって、コマンド(12バイト)の内容が規格に準拠しているかの可否判断が行なわれる。これらの判別結果が「可」とされた場合、「メディアチェック」によって、再生装置70に装填されているディスク90が、例えばオーディオCDであるか否か(つまり、CD-ROMなどではなくCD-TEXTとしてテキストデータを有する可能性のあるディスクであるか否か)の判別が行なわれる。なお、CD-TEXTとは種別的にはオーディオCD(CD-DA)に含まれるものである。ディスク90がオーディオCD(オーディオトラックが存在する)か否かは、上述したTOCにより判別できる。そして通常、TOCデータはディスク装填時に読み込んでいるため、即座に判別可能である。

【0074】ここで、ディスク90がオーディオCDであると判別された場合は、「Allocation length=0」か否か、即ち、要求するデータ長の確認を行ない(S003)、要求するデータがある場合40 (「0」でない場合)ステップS004に進む。なお、各種チェック処理(S002)における判定が「否」であった場合は、その時点で所要のエラー処理が行なわれ、バファリング処理は行なわれないことになる。

【0075】要求するデータ長の確認(S003)が終了すると、ファームウェア内の状況の初期化を行なう(S004)。そして、例えば現在当該再生装置70によってオーディオ再生が行なわれている場合は、その再生処理を50 強制的に終了する(S005)。但し、現在オーディオ再生

処理を行なっていない場合は、この処理行程は省略される。ステップS005を経ると、さらにキャッシングパラメータの初期化を行なう(S006)。これは、キャッシングメモリ20の使用形態が、通常の状態から図15に示した状態(ローサブコードエリアとローテキストエリアがマッピングされる状態)に移行するため、今まで格納されているデータが不要となるので、これらのデータを無効にするための処理とされる。そして、本実施の形態のバファリング処理を行なうために実行すべき割り込み処理の切替えを行なう(S007)。これは、例えばディンターリープをオフにするなど、バファリング処理を行なうための所要の割り込み処理を実行させるためとされる。

【0076】図16のフローチャートにおいて割り込み処理の切替え処理(S007)を経ると、図17のフローチャートに示されていうように、ディスク90のリードインエリアのTOCに対するアクセス要求を行なう(S008)。これによりピックアップ1はディスク90の内周側に記憶されているTOCに対してアクセスを開始する。さらに、ローサブコードデータが96バイト分バファリングされた時点で「1」づつインクリメントされるパラメータとして設定される【N】、テキストグループの先頭を示すヘッダ情報とされるスタートヘッダが検出された回数を示すパラメータとして設定される【M】、及びCRCのNG回数を示すパラメータとして設定される【NG数】を初期値化「0」する(S009)。アクセス開始後は、アクセス完了に対して待機状態となり(S010)、アクセスが完了した時点で、ローサブコードデータのバッファリングを開始する(S011)。即ち、ディンターリープをオフにするとともに、図15に示したローサブコードエリアに対してローサブコードの格納を開始させる。

【0077】そして、ローサブコードエリアに対して所定量のローサブコードデータ(例えば1サブコードに相当するデータ量)が格納されるのを待つことになる(S012)。ここで、所定量のローサブコードデータが格納されると、その時点で割り込み処理として所要の通知処理が行なわれ、システムコントローラ10ではローサブコードエリアに対してローサブコードデータが格納されたことを認識することができる。なお、この図に示すフローチャートとしてはステップS012の判別処理を便宜上一連の処理行程の一部として示しているが、実際には、この判別処理は随時行なわれているものとされる。つまり、ローサブコードエリアに対するローサブコードデータの読み込みも、随時行なわれている。したがって、以降説明する処理行程は、ローサブコードエリアに所定量のローサブコードデータが格納される度に、随時割り込み処理として行なわれるようになれる。

【0078】ローサブコードエリアに所定量のローサブコードデータが蓄えられると、これをカウントするパラメータ【N】をインクリメント($N=N+1$)する処理

を行なう(S013)。そして、蓄えられたローサブコードデータに対してCRCのチェックを行なう(S014)。このCRCチェックは、図8に示した4個のCRC領域に或る誤り検出符号に対して行われ、チェックによる判定結果としては、4個のCRC領域全てが「OK」とされた場合に「OK」とされ、一つでも「NG」があった場合には「NG」とされる。ステップS014で、CRCが「OK」となった場合、即ちローサブコードエリアに蓄えられたローサブコードデータが有効であるとされた場合は、テキストグループの先頭を示すスタートヘッダのパターン(Start_Header_Pattern)のチェックを行なう(S016)。

【0079】そして、スタートヘッダの検出の有無の判別を行ない(S017)、スタートヘッダが検出されない場合は、ステップS012に戻る。また、スタートヘッダが検出された場合は、検出されたスタートヘッダの数をカウントするパラメータ【M】をインクリメント($M=M+1$)する処理を行ない(S018)、続いて、パラメータ【M】の値の判別を行なう(S019)。ここで、パラメータ【M】が「1」以下であると判別した場合は、【Count1】にパラメータ【N】の値を設定して(S020)、ステップS012に戻る。なお、【Count1】は、何回目のローサブコードデータの読み込みにおいて第一のスタートヘッダが検出されたかを示すパラメータとされる。

【0080】一方、ステップS019においてパラメータ【M】が「2」または「2」以上であると判別した場合は、【Count2】にパラメータ【N】の値を設定する(S021)。この【Count2】は、何回目のローサブコードデータの読み込みにおいて第二のスタートヘッダが検出されたかを示すパラメータとされる。即ち、パラメータ【M】が「2」または「2」以上となった時点で、第一のスタートヘッダ及び、第二のスタートヘッダが検出されたことになる。つまり、第一、第二のスタートヘッダ間に1個のテキストグループがバッファリングされたとみなすことができる。

【0081】以降、このようにして検出された第一、第二のスタートヘッダ間におけるテキストデータの有効性を検証する処理(2)に移行するが、これについては後で図21にしたがい詳しく説明する。また、ステップS021を経て処理(2)に移行した場合でも、図17に示したバッファリング処理は継続して行なうようにしている。これにより、後述する処理(2)においてテキストデータの有効性が確認されなかった場合でも、その時点で引き続きキャッシングメモリ20のローサブコードエリアに格納されているローサブコードデータの検証を行なえば良い。したがって、改めてTOCに対するアクセス処理などを行なう必要がないので、処理効率が向上する。

【0082】また、例えば2個のスタートヘッダが検出された時点でローサブコードデータのバッファリングを

終了させたうえで、テキストデータの有効性の検証処理に移行するようにしても良い。これは、2個のスタートヘッダが検出されたことによって、その時点で期待されるテキストデータが得られたと見なすようにする処理とされる。これにより、2個のタートヘッダが検出された後は、システムコントローラ10は前記検証処理のみを実行されれば良いので、処理負担を軽減することが可能である。

【0083】ところで、ステップS014においてCRCが「NG」となった場合は、CRCのNG回数を示すパラメータとして設定される「NG数」のインクリメント（NG数=NG数+1）を行ない（S022）、さらに、「NG数」が当該再生装置70に設定されている「NG許容数」（例えば「4」回）以下であるか否かの判別を行なう（S023）。ここで、「NG数」が「NG許容数」以下であった場合は、リトライ処理としてステップS012に戻り、引き続きサブコードデータが読み込まれるのを待つ。また、「NG数」が「NG許容数」以上であった場合は、リトライ処理としてステップS008に戻りTOCに対するアクセス要求を行ない、さらに上述したステップS009からの処理を実行していくようとする。

【0084】なお、本例では「NG許容数」は例えば「4」として、「NG数」がこれを超えた場合に再びTOCに対するアクセス要求を行なうようにしている。しかし、「NG数」が例えば256（全てのパックについて「NG」の判定）を超えるような場合、テキストデータを読むことができる見込みがないものと見なして、図16、図17に示すフローチャートの処理自体を終了させるようにしても良い。

【0085】この場合の処理行程としては、ステップS014、S022を経た後、例えば図18に示されているようになる。この図18において、図17と同一の符号を付した処理ステップについては同じ処理行程を示している。図示されているように、ステップS022において「NG数」のインクリメント処理を経ると、パラメータ「M」、即ち、テキストグループの先頭を示すヘッダ情報とされるスタートヘッダが検出された回数が「0」であるか否かの判別を行う（S701）。そして判別結果が「0」とされスタートヘッダが検出されないと判別した場合は、ローサブコードデータが例えば96バイトを単位として何回バッファリングされたかを示すパラメータ「N」が所定値以上であるか否かの判別を行う（S702）。なお、ステップS702における所定値とは、ディスク90にテキストデータが記憶されているか否かを判別するための閾値として設定される。即ち、ステップS014においてCRCが「NG」になった時点で、ローサブコードデータが何回バッファリングされたかに基づいて、以降の処理が行われる。

【0086】例えば、パラメータ「N」が所定値以下であったと判別した場合は、さらに、「NG数」が例えば

「64」を超えたか否かの判別を行う（S703）。このステップS703における閾値とされる「64」は、先述したようにテキストグループを形成する1ブロックが256パック（256パック=64サブコードフレーム）によって形成されていることに基づいている。したがってステップS703において「NG数」が例えば「64」を超えていたと判別した場合は、再生装置70に装填されたディスク90にはテキストデータが格納されていないと判断する（S704）。これにより、テキストデータの読み込むことができる見込みがないものと見なして、バッファリング処理を終了させる（S705）。また、「NG数」が例えば「64」を超えていないと判別した場合は、リトライ処理としてステップS012に戻り、引き続きサブコードデータが読み込まれるのを待つ。

【0087】また、ステップS702においてパラメータ「N」が所定値を超えていなかった場合には、「NG数」が当該再生装置70に設定されている「NG許容数」（例えば「4」回）以下であるか否かの判別を行なう（S023）。ここで、「NG数」が「NG許容数」以下であった場合は、リトライ処理としてステップS012に戻り、引き続きサブコードデータが読み込まれるのを待つ。また、「NG数」が「NG許容数」以上であった場合は、リトライ処理としてステップS008に戻りTOCに対するアクセス要求を行ない、図17で説明した場合と同様にステップS009からの処理を実行していくようする。

【0088】このように、ステップS701、S702、S703を経ることによって、スタートヘッダが検出されず、サブコードデータのバッファリング回数が所定値以下であり、かつCRCの「NG数」が例えば「64」以上であった時点で、ディスク90にテキストデータが格納されていないことを判別することができるようになる。したがって、ディスク90がオーディオCDとしてテキストデータを有していないと判別した時点で、図16、図17に示したバッファリング処理を終了させることができる。つまり、再生装置70にテキストデータが記憶されていないオーディオCDが装填されている場合は、バッファリング処理が開始されると、連続してCRCがNGとされる処理ステップを進むことになる。この場合、スタートヘッダは検出されないので、バッファリングの回数（S702）及び「NG数」（S703）に基づいて、当該ディスク90にテキストデータが記録されているか否かを判別することができるようになる。これにより、バッファリング処理が開始されてから、必要最小限のサブコードデータのバッファリングを行うことで、テキストデータの有無を判別することができる。

【0089】ところで、ステップS014においてCRCが「NG」とされた場合は、図17において例えば破線で示されている結合子（3）に対応した、図19に示されているフローチャートの処理を行なうことも考えられ

る。この図に示されているフローチャートの処理行程としては、ステップS201に示されてるようにパラメータ『M』が「0」であるか否かの判別を行ない、ここでパラメータ『M』が「0」であると判別した場合には、図17に示したステップS023に進み『NG数』の判定を行なうようにする。つまり、ステップS201でパラメータ『M』が「0」であった場合、まだスタートヘッダが検出されていない状態を示しており、したがって、再度スタートヘッダを検出する処理を行なうようにするために、ステップS023におけるNG判定処理を経ることになる。

【0090】また、パラメータ『M』が「0」ではないと判別した場合は、ステップS202に進みパラメータ『M』が「2」であるか否かの判別を行なう。ここで、パラメータ『M』が「2」ではないと判別した場合は、図17に示したステップS008に進みTOCエリアに対するアクセス要求を行なう。一方、パラメータ『M』が「2」であると判別した場合は、サブコードデータのバッファリング処理を終了して(S203)テキストデータの有効性を検証する処理(2)に進む。

【0091】ステップ201及びステップS202の行程により、検出されているスタートヘッダの連続性が検出されるようになる。但し、図17に示したフローチャートのステップS019において、パラメータ『M』が例えば「2」または「2」以上とされた場合は、ステップS012を経てテキストデータの有効性を検証する処理(2)に進む行程を説明している。したがって、パラメータ『M』が「2」または「2」以上とされている場合、スタートヘッダの検出に伴って実行される処理行程として、図17のステップS019、S021を経て処理(2)に進むことになる。

【0092】しかしながら、前述したようにステップS012以降の処理行程は隨時行なわれており、現在実行している各判別処理などに対して並行した読み込み処理が行なわれているものとされる。これは、処理中のデータに対して予め次のデータの読み込みを行なう先読み処理に相当する。したがって、ステップS014においてCRCの判定結果が「NG」とされた場合でも、ステップ201及びステップS202の処理行程において先読みによって連続したスタートヘッダが検出された状態では、当該スタートヘッダに対応したテキストデータに対して有効性の検証を行なうことが可能とされるようにしている。なお、図19のフローチャートにおいて、テキストデータの有効性の検証を行なうまえにステップS203においてローサブコードデータのバッファリングを終了しているのは、CRC判定が「NG」とされていることに加え、先読みによって得られたテキストデータを利用すれば良いということから、これ以上バッファリング処理を継続しなくても良いという判断を下すことが可能とされるためである。

【0093】図20はバッファリング処理によるキャッシュ

ユメモリの内容を説明する図であり、図20(a)は、図17のフローチャートで説明した各パラメータの数値例、図20(b)は、キャッシュメモリ20のアドレスに準じたデータ内容を模式的に示す図である。

【0094】パラメータ『N』は、1フレーム分のサブコードデータがキャッシュメモリ20のサブコードエリア格納された場合にインクリメントされ、この図に示す例では初期値とされる「0」から「517」までの数値が示されている。即ち、図17におけるステップS012以後の処理が517回行なわれたことを示している。これに対応して、図20(b)に示すキャッシュメモリ20では、Cash#1～Cash#517が示されているが、パラメータ『N』に対応したエリアにはバッファリングされた1フレーム分のサブコードデータが格納されている。

【0095】パラメータ『M』は、スタートヘッダが検出された時点でインクリメントされ、この図に示す例では『N』=「7」、及び『N』=「516」となった場合に、インクリメントされている。したがって、『N』=「7」の時点で第一のスタートヘッダが、また『N』=「516」の時点で第二のスタートヘッダが検出されたことを示している。『NG数』は『N』=「1」の時点からインクリメントされ、『N』=「4」以降は『NG数』=「4」がホールドされている。即ち、『N』=「5」以降はCRC判定が「OK」とされていることを示している。

【0096】『Count1』はパラメータ『M』が1回目にインクリメントされた時点でのパラメータ『N』の値が設定され、本例では「7」が設定されている。同様に『Count2』はパラメータ『M』が2回目にインクリメントされた時点で、パラメータ『N』の値が設定され「516」が設定されている。

【0097】このようなことから、図20(b)に示されているように、キャッシュメモリ20のローサブコードエリアでは、Cash#1～Cash#4がCRC判定により「NG」とされたデータとされ、『Count1』=「7」～『Count2』=「516」に相当する、Cash#7～Cash#516までの間にテキストグループが存在すると判別することができる。即ち、これまでの処理が、図16乃至図19で示したフローチャートの処理行程によって行なわれる。なお、Cash#7～Cash#516については、後に8ビット／6ビット変換されキャッシュメモリ20におけるローテキストエリア内に移されることになるが、この処理については後述で詳しく説明することにする。

【0098】図17のフローチャートのステップS016乃至ステップS017においてスタートヘッダを検出する行程を説明したが、ここではその概要を説明する。例えば15024フレーム分のローサブコードデータをバッファリ

ングすることによって、その中からスタートヘッダを検出することも考えられるが、CD-TEXTとして記憶されているデータ容量が比較的小さい場合などは、処理効率が良いものではない。そこで、本例では、ローサブコードデータをローサブコードエリアに格納しつつ、並行して格納されているスタートヘッダの監視を行い、検出されたスタートヘッダの数が2以上である場合、その間に1個のテキストグループがあるという判断を下している。

【0099】スタートヘッダのパターンとしては「80h, 00h, 00h, X0h」とされる4バイトのデータとされる。このデータ配列をビットで示すと

「80h」 . . . 10000000b

「00h」 . . . 00000000b

となる。なお4バイト目の「X0h」については、「00h」または「80h」のいずれかであることを示している。つまり、このヘッダパターンは「80h, 00h」から、図13で説明したようにアルバムタイトルヘッダであることがわかる。

【0100】ローサブコードデータがローサブコードエリアに格納された状態では図21に示されているようになる。この図には、スタートヘッダを含んだP～Wに至る8ビットのデータを示している。このデータにおいて、図7(a)で説明したパックの先頭(ヘッダ領域)をチェックし、これらのパターンが2回以上検出された場合に、1個のテキストグループが検出されたとするとができる。

【0101】6. 8ビット／6ビット変換

ローサブコードエリアに格納されたローサブコードは図21に示したように8ビット(P～W)とされているが、テキストデータは6ビット(R～W)のデータとされる。したがって、効率の良い処理を行なうためには、8ビットのデータ(P～W)のうちP、Qデータの2バイトを除去することが必要になる。そこで、後で図23で説明するフローチャートの処理行程とされる8ビット／6ビット変換処理を行なうようにしている。

【0102】図22(a)は、図21に示したR～Wのデータ配列を、アルファベット(a～x)に対応させて示している。この状態ではR～Wのデータは6ビット単位とされる。このように配列されている8ビットのデータからP、Qデータ(X印)を除去して再び8ビットに配列し直すと、図22(b)に示されているようになる。つまり、キャッシングメモリ20のローテキストエリアには、バイト単位で扱うことができるテキストデータが格納される。

【0103】この処理について、図20を参照して説明する。図20(b)に示されている「Count1」～「Count2」とされるテキストグループを含んだデータは、図22(a)に対応した8ビット(P～W)のデータとされる。つまり、ローサブコードエリアには、

不要なP、Qデータも格納されている。そこで、8ビット／6ビット変換を行なうことによって、「Count1」～「Count2」におけるテキストグループのみをローテキストエリアに移して、該ローテキストエリアにおいて、後述する例えばトラックタイトルヘッダやサイズインフォメーションヘッダの検証処理を行なうようしている。

【0104】ローサブコードエリアの他にローテキストエリアを設けるようにした場合、キャッシングメモリ20

10 内に必要な領域が増えることになる。しかし、検証処理などを行なう場合、R～Wデータの6ビット毎の検証を行なうよりも、バイト単位で検証を行なう方が、システムコントローラ10の処理も単純化することができ、さらには、ホストコンピュータ80に対する転送処理も簡単になる。

【0105】7. テキストデータ有効性検証処理

図23、図24は、図16、図17で説明したバッファリング処理に続くテキストデータ有効性検証処理の流れを説明するフローチャートを示す図である。なお、図2

20 3に示されているフローチャートの結合子(4)及び結合子(5)は、それぞれ図24に示されているフローチャートの結合子(4)及び結合子(5)に対応したものである。即ち、テキストデータ有効性検証処理としては図23乃至図24に示されている一連の処理行程によって実現される処理とされる。

【0106】テキストデータ有効性検証処理としては、まず図22で説明した8ビット／6ビット変換を行な

い、テキストデータに対応した6ビット(R～W)のデータをローテキストエリアに格納していく(S301)。そして、パックのヘッダ領域の各ID(ID1～ID4)を示すためのパラメータ「U」に「0」を設定し、さらにパラメータ「M」に「0」を設定する行なう(S302)。

【0107】ところで、ステップS301における8ビット／6ビット変換が開始されると、ローテキストデータがローテキストエリアに順次格納されていくことになるが、ここで順次格納されてくるデータに対してスタートヘッダ(先頭パックのヘッダ領域)のサーチを行なう。スタートヘッダのパターンは、上述したように例えば

40 「80h, 00h, 00h, X0h」とされている。これらのパターンはそれぞれ、図9(b)に示したヘッダ領域のID1～ID4に相当する。したがって、格納されてくるローテキストデータにこのパターンが含まれているか否かを判別することで、スタートヘッダを認識することができる。

【0108】スタートヘッダのサーチ処理としては、先頭のU番目のデータ(ID1に相当する)が10000000bか否か(S303)、U+1番目(ID2に相当する)のデータが00000000bか否か(S304)、U+2番目(ID3に相当する)のデータが00000000

50

0 0 bか否か(S305)、さらにU+3番目(ID4に相当する)のデータが0 0 0 0 0 0 0 bまたは1 0 0 0 0 0 bか否か(S306)の判別を行なう。

【0109】ここでの判定結果として、いずれか一つでも一致しない場合は、パラメータ『U』のインクリメントを行ない(S313)再びステップS303からステップS306におけるスタートヘッダサーチを行なう。このようにパラメータ『U』をインクリメントすることにより、前回1番目のデータをID1に相当するものとしていたが、今回は前回の2番目のデータ、即ち前回ID2に相当するとしていたデータを、ID1としてサーチ処理を行なう。このように、被サーチデータをシフトさせていくことで、ローサブコードデータの中から所望するスタートヘッダパターンを検出するようにしている。

【0110】ステップS303からステップS306において、全てのデータのパターンがスタートヘッダパターンに一致すると、パラメータ『M』をインクリメントする(S307)。即ちパラメータ『M』は先程と同様にスタートヘッダの検出回数を示すものとして用いられている。最初にステップ(S307)に到達した時点では『M』=1とされ、これは第一のスタートヘッダが検出されたことを示す。したがって、ステップS308における『M』の値の判定は「NO」とされ、スターへッダに続く2番目のパックのヘッダパターンの検出処理に移行する(S309)。

【0111】ここで、2番目のパックのヘッダパターンの検出処理について説明する。本例では、スタートヘッダに続くヘッダのタイプとして、例えばトラックタイトルヘッダまたはアルバムタイトルヘッダの検証を行なうこととしている。図25はCD-TEXTフォーマットに規定されているヘッダタイプの一例を示している。即ち、図13の一部を詳細に示した図である。図中(a)に示されている部分がアルバムタイトルヘッダとされており、先頭のID1が「80h」、続くID2は「00h」とされる。したがって、少なくともID1、ID2がこのようなパターンとされていた場合は、当該ヘッダはアルバムタイトルヘッダとすることができます。

【0112】また、図中(b)で示されている部分がトラックタイトルヘッダとされており、先頭のID1が「80h」とされ、続くID2については曲順を示す「01h」「02h」・・・「63h」とされている。即ち、少なくともID1及びID2がこれらの値とされているパターン(例えば1曲目を示す「80h, 01h」など)はトラックタイトルヘッダと見なすことができる。そして、後述するようにアルバムタイトルヘッダ、またはトラックタイトルヘッダはパラメータ『M』=1とされたスタートヘッダが検出された次(順方向)のパックの先頭、即ち18バイト後から始まるテキストグループ内の2番目のパックのヘッダパターンとされる。したがって、第一のスタートヘッダとされているテキストグループ内の先頭パックのヘッダ及びこれに続く

2番目のパックのヘッダとされる、2個のヘッダの内容によって第一のスタートヘッダが認識されるようにしている。

【0113】したがって、2番目のパックのヘッダパターンを検出する場合のフローチャートは図26に示されているようになる。但し、この図に示すフローチャートは、図23のステップS309の処理内容を示している。ステップS401において、まず次のパックの先頭とされるU+18番目のデータ(ID1に相当する)と、これに続くU+19番目のデータが「80h, 01h」に一致しているか否かを判別し、各データがそれぞれ「80h, 01h」に一致していると判別した場合は、所要のヘッダ(トラックタイトルヘッダ)が検出されたと見なす(S402)。

【0114】また、各データが「80h, 01h」に一致していない場合は、U+18番目のデータ、U+19番目のデータ、及びこれに続くU+20番目のデータ

(ID3に相当する)が、「80h, 00h, 01h」に一致しているか否かを判別する(S402)。ここで判別するヘッダタイプは、ID1、ID2による「80h, 00h」からアルバムタイトルヘッダとされる。また、ID3に示されている「01h」は上述したようにシーケンスナンバーが示されている。これは後述する図29に示されているが、先頭パックが「00h」、2番目のパックが「01h」・・・というように、ブロック内における当該パックの順位を示すことができるようになっている。

【0115】したがって、先頭パックのヘッダ(スタートヘッダ)としては「80h, 00h, 00h」とされる。そして、記憶されているアルバムタイトルが比較的長いものとされ、例えば先頭パックと2番目のパックにまたがって記憶されている場合は、「80h, 00h, 01h」となる。したがって、2番目のパックの先頭はID1、ID2によるトラックタイトルヘッダ「80h, 01h」、またはID1、ID2、及びID3によるアルバムタイトルヘッダ「80h, 00h, 01h」となる。したがって、各ヘッダがそれぞれ「80h, 00h, 01h」に一致している判別した場合は、所要のヘッダ(アルバムタイトルヘッダ)が検出されたと見なし(S403)、また、一致していないと判別した場合は所要のヘッダが検出されなかつたとみなす(S404)。

【0116】なお、パックのヘッダ領域はID1~ID4によって形成されるが、これら全てのデータを検出する必要はなく、説明したように少なくとも、ヘッダの判別が可能とされるIDの検出を行なうことができれば良い。また、この図のフローチャートでは例えば、例えばトラックタイトルヘッダ、アルバムタイトルヘッダの検出を行なう例を挙げているが、先頭パックに続く2番目のパックのヘッダであることを検出することができれば、検出項目はいずれのものであっても良い。

【0117】図23のフローチャートに戻り説明を続ける。ステップS309を経た後、所要のヘッダが検出されたか否かを判別する(S310)。そして、所要のヘッダが検出された場合は、システムコントローラ10はU番目のデータが格納されているローテキストエリアのアドレスをスタートアドレス(Start_Address)であると認識して記憶する(S311)。そして、ステップS313によってパラメータ『U』のインクリメントを行ない、再度スタートヘッダのサーチ処理を行なう。

【0118】また、ステップS310で所要のヘッダが検出されなかったと判別した場合は、パラメータ『M』のインクリメントを行ない(S312)、パラメータ『M』を元の値に戻す。これは、第一のスタートヘッダについて、所要のヘッダが検出されなかったという結果に基づいて、再び第一のスタートヘッダのサーチを行なうためである。したがって、第一のスタートヘッダに対して所要のヘッダが検出されるまで、ステップS308からステップS309に進むようにされる。そして第一のスタートヘッダに対して所要のヘッダが検出され、パラメータ『M』が「2」となった時点で、結合子(4)に対応した処理に進む。

【0119】パラメータ『M』が「2」となると図24に示されているように、システムコントローラ10はU番目のデータが格納されているローテキストエリアのアドレスをエンドアドレス(End_Address)であると認識して記憶する(S314)。そして、エンドアドレスの値とスタートアドレスの値の差を求め、この差、即ち各アドレス間のデータ長と、テキストデータの最小単位として規定されているパックの長さ(Pack_Length...18バイト)と比較する(S315)。ここでアドレス間のデータ長がパックの長さよりも長いと判別した場合は、次に、前記各アドレス間のデータ長が、1ブロックが最大256パックで形成された場合の最大データ長(Max.TEXT_Length...36.864Kバイト)以下であるか否かを判別する(S316)。そして、前記各アドレス間のデータ長が最大データ長以下であると判別した場合は、サイズインフォメーションヘッダのチェックを行なう(S319)。つまり、ステップS315、S316を経ることによって第一のスタートヘッダに続く第二のスタートヘッダが存在する見なすことができる。そして、さらに第二のスタートヘッダの存在を確認するために、この第二のスタートヘッダから所定量越った位置に在る、サイズインフォメーションヘッダのチェック(S319)に移行する。

【0120】なお、ステップS315において各アドレス間のデータ長がパックよりも短い場合や、またステップS316において各アドレス間のデータ長が前記最大データ長よりも長いと判別した場合、システムコントローラ10はステップS317に進み、エンドアドレスをヘッダの起点であると認識して記憶し、パラメータ『M』を初期値化

「0」する処理を行なう(S318)。これは、第二のスタートヘッダが検出されなかつたという結果に基づいて、再び第一のスタートヘッダのサーチを行なうためである。したがって、パラメータ『M』を「0」にした後は、結合子(5)を介して図23に示すステップS303に進む。

【0121】この処理においては、図25(c)に示したサイズインフォメーションヘッダ#1～#3の検出を行なう。ところで、サイズインフォメーションヘッダ

10 (#1～#3)としては、図25(c)に示されているように先頭のID1が「8Fh」とされ、続くID2はそれぞれ「00h」「01h」「02h」とされている。即ち、少なくともID1及びID2がこれらの値とされているパターンはサイズインフォメーションヘッダと見なすことができる。また、このサイズインフォメーションは、テキストグループの最後の3個のパックのヘッダとされているので、本例では第二のスタートヘッダから遡って検出するようにしている。

【0122】したがって、サイズインフォメーションヘッダチェック(S319)の処理としては、例えば図27のフローチャートに示されているようにされる。まずステップS501において、最後のパックの先頭とされるU-18番目のデータ(ID1に相当する)と、これに続くU-17番目のデータ(ID2に相当する)が「8Fh, 02h」に一致しているか否かを判別し、各データがそれぞれ「8Fh, 02h」に一致していると判別した場合はステップS502に進む。ステップS502では、最後から2番目のパックの先頭とされるU-36番目のデータ(ID1に相当する)と、これに続くU-35番目のデータ(ID2に相当する)が「8Fh, 01h」に一致しているか否かを判別し、各データがそれぞれ「8Fh, 01h」に一致していると判別した場合はステップS503に進む。そして、ステップS503では、最後から3番目のパックの先頭とされるU-54番目のデータ(ID1に相当する)と、これに続くU-53番目のデータ(ID2に相当する)が「8Fh, 00h」に一致しているか否かを判別する。

【0123】ステップS501からステップS503を経ることにより、各データがそれぞれ所望するデータ内容に一致していた場合はステップS504に進み、3個のサイズインフォメーションヘッダが検出されたと見なす。また、ステップS501からステップS503において、一つでも内容が一致しないものが在れば、不一致が検出された時点でステップS505に進みサイズインフォメーションヘッダが検出されていないと見なす。

【0124】このようにして得られたサイズインフォメーションの検出結果に基づいて、図24に示すステップS319以降の処理を行なう。ステップS319を経た後は、まずサイズインフォメーションヘッダが検出されたか否かの検出を行ない(S320)、ここで、サイズインフォメー

ションヘッダが検出された場合は、所望するテキストデータが得られたと見なし、図16、図17によって開始されているバファリング処理を終了して(S321)、ステップS311で認識したスタートアドレスとステップS314で認識したエンドアドレスの間のテキストデータをホストコンピュータ80に転送する(S322)。

【0125】再生装置70としては、上記したような処理によりテキストデータの誤検出を低減することができる、ホストコンピュータ80に対しては精度の高いテキストデータを転送することができる。そして、ホストコンピュータ80では、ディスク90から読み出された精度の良いテキストデータを利用して所要の表示処理を行なうことができるようになる。

【0126】8. データ例

図28はバファリング処理を経て、キャッシュメモリ20のローサブコードエリアに格納されたローサブコードデータの一例を示す図である。この図に示すローサブコードデータは、CRCのNG数が例えば9回、バファリングフレーム数(パラメータ『N』)が例えば100回とされるデータである。したがって、バファリングされたデータの先頭から $9 \times 96 = 864$ バイトは無効とされる。即ち、アドレス0から863(deci)までのデータは図20(b)に示したCRC_NG領域に相当する。したがって、9フレーム以降からCRCが「OK」とされている。また、スタートヘッダとしては864番目からP、Qデータを含んだ第一のスタートヘッダパターン(20h, 00h, 00h, 00h...実線で囲む)が示され、これは例えば「Count1」に対応したデータとされる。そして、1944番目から同じくP、Qデータを含んだ第二のスタートヘッダパターン(20h, 00h, 00h, 00h)が示されているが、これが「Count2」に対応したデータとされる。したがって、この第一、第二のスタートヘッダの間に、所望するテキストデータが存在していると見なし、テキストデータ有効性検証処理を行なう。

【0127】図29はバファリング処理、テキストデータ有効性検証処理を経て、ローテキストエリアからホストコンピュータ80に転送されるローテキストデータの一例を示す図である。この図のデータの先頭に実線で囲んだスタートヘッダに相当するアルバムタイトルヘッダ「80h, 00h, 00h, 00h」が示され、このスタートヘッダパターンの次に、下線によりアルバムタイトルヘッダ「80h, 00h, 01h」が示されていることが解る。これは、上記したようにスタートヘッダパターンから数えて18バイト後に位置している。したがって、実線で囲んだ先頭からのデータが第一のスタートヘッダパターンとすることができる。

【0128】なお、この図に示すデータ例では、アルバムタイトルが2つのパックにまたがっているので、1曲目のトラックタイトルヘッダ「80h, 01h」は3番

目のパックのヘッダとされる。したがって、ID3のシーケンスナンバは「02h」とされている。つまり、当該ディスク90のアルバムタイトルが1パックに収まる比較的短い場合は、2番目のパックが1曲目のタイトルに対応し、ID1～ID3までの内容は、「80h, 01h, 01h」となる。

【0129】また同じく実線で囲んだ811番目から実線で囲んだスタートヘッダパターンが示され、このスタートヘッダパターンを所定量越った位置に、サイズインフォメーションヘッダパターン(「8Fh, 02h」、「8Fh, 01h」、「8F, 00h」)が示されていることが解る。したがって、811番目からのデータが第二のスタートヘッダパターンとすることができる。

【0130】したがって、これら2個のスタートヘッダに囲まれたデータが、所望するテキストデータとされる。即ち、図29に示す例では、各スタートヘッダ(「80h, 00h, 00h, 00h」)の間のデータが、ローテキストエリアからホストコンピュータ80に転送されるようになる。

【0131】9. 変形例

以下、本発明の変形例を説明する。上記実施の形態では、例えば2個の連続したスタートヘッダを検出して、各スタートヘッダの間の所定のデータ内容の検証を行なう例としたが、例えば図30の模式図に示されているように2個以上のスタートヘッダH1、H2、H3...を検出して、これらのスタートヘッダがCD-TEXTのフォーマットとして規定されているパターンであった場合にテキストデータの有効性を確認するようにしても良い。また、例えば2個以上のスタートヘッダを検出した場合、スタートヘッダH1、H2に囲まれたデータの内容(テキストデータ(1))と、スタートヘッダH2、H3に囲まれたデータの内容(テキストデータ(2))を比較して、双方のデータ内容が一致していることを確認した場合には、テキストデータ((1)及び(2))の有効性を確認するようにしても良い。

【0132】また、上記実施の形態ではバファリング処理によって検出された2個のスタートヘッダの間に在る、アルバムタイトルヘッダ、トラックタイトルヘッダやサイズインフォメーションヘッダの内容が所定のものとされていれば、当該トラックタイトルヘッダやサイズインフォメーションヘッダに対応したテキストデータの有効性を確認したが、他のヘッダ内容によても有効性の確認を行なうことができる。

【0133】例えば、ヘッダ(図29の実線で囲んだ部分に相当する)の内容チェックとして、最初の1バイト(ID1:Pack_Type_Indicator)をチェックする場合を例に挙げる。ここで、第一のスタートヘッダがU番目にバファリングされ、第二のスタートヘッダがU+N番目に在るとする。そして、検証対象とする任意のヘッダがU+M番目(但し、M<N)とし

た場合、処理の流れとしては図31のフローチャートに示されているようになる。

【0134】図31のに示されているように、まず、ID1が「80h」であるか否かを判別する(S601)。ここで、ID1が「80h」であると判別した場合は、さらにID2が「0」以上かつ「63」以下であるか否かを判別する(S602)。そして、ID2が「0」以上かつ「63」以下であったと判別した場合は、図13に示したように、アルバムタイトルヘッダまたはトラックタイトルヘッダであると見なすことができ、テキストデータの有効性を確認することができる(S615)。また、ID2が「0」以上かつ「63」以下ではないと判別した場合、即ちCD-TEXTのフォーマットで規定されている値ではない場合には、テキストデータとしての有効性を確認しないで、無効なデータであると認識する(S616)。

【0135】以降、同様にして、ステップS603からステップS614に至るまで、各ステップ毎にID1の内容が「81h」から「8Fh」のいずれかに一致しているか否かの判別を行なう。なお、「81h」から「8Fh」についてのデータ内容(識別内容)は、図13に示したものとされる。このようにしてステップS603～ステップS614においての判別結果が、CD-TEXTのフォーマットで規定されている値に一致していない場合は、ID1に対応したテキストデータは無効なデータであると認識する(S617)。一方、ステップS603～ステップS614において、一つでも「81h」から「8Fh」に一致しているものがあったと判別した場合には、その時点でテキストデータの有効性を確認することができる(S615)。なお、この図に示すフローチャートでは、一つの任意のヘッダとして例えばID1(Pack_Type_Indicator)のチェックを行なう例を挙げたが、他のヘッダ(ID2, ID3, ID4)について同様の処理を行なうようにしても良い。

【0136】また、ローサブコードエリアに対してローサブコードデータをバッファリングした時点で、パックの先頭のIDを全てスタートヘッダ(テキストグループの先頭のヘッダ)であるか否かを検出することができるハードウエアを構成することにより、システムコントローラ10の処理負担が軽減され、バッファリングを行なうデータ量を必要最小限に低減することができる。

【0137】なお、上記実施の形態は、テキストデータがディスク90のTOC領域に記憶されている「モード4」を例に挙げて説明したが、テキストデータがプログラムエリア(主データのエリア)に記憶されている「モード2」に対しても適用することができる。

【0138】

【発明の効果】以上、説明したように本発明は、例えばオーディオCDの一種とされるCD-TEXTに記憶さ

れている、テキストデータ(文字情報)を構成するテキストグループの先頭にあるスタートヘッダ(先頭パックのヘッダ)を検出するようにしている。これにより、当該テキストグループの開始位置、終了位置を検出することができる。そして、前記スタートヘッダ間にある所定のデータ内容の検証や比較を行なうことにより、前記スタートヘッダの間のデータに対して、テキストデータとしての有効性を確認することができ、テキストデータの誤検出を低減することができるようになる。

- 10 【0139】例えば、前記スタートヘッダが検出されるとパックが形成されていることを想定することができる。つまり、前記スタートヘッダの間に在るデータに対して、パックのヘッダに相当するデータ内容を検証し、この検証結果に基づいて前記スタートヘッダ間のデータを前記テキストデータとして認識することができるようになる。したがって、第一のスタートヘッダから順方向における所定の位置にあるパックのヘッダ(例えばアルバムタイトルヘッダ、トラックタイトルヘッダ)と、第二のスタートヘッダから逆方向における所定の位置にあるパックのヘッダ(例えばサイズインフォメーションヘッダ)のパターンを検出することにより、テキストデータの有効性を確認することができる。つまり、前記スタートヘッダ間の所要の位置にCD-TEXTフォーマットに規定されているパックのヘッダに対応したデータパターンを検出することにより、テキストデータの有効性確認が可能となる。
- 20 【0140】また、スタートヘッダ間における例えばヘッダ(ID1...パックインジケータ)などの所定の位置にあるデータ内容が、規定されている所定のデータ内容に一致している場合にも、テキストデータの有効性を確認することができる。つまり、読み込まれたデータと規定されているデータの内容の一致、不一致を判定することにより検証結果を得るので、テキストデータの有効性確認の精度を向上することができる。
- 30 【0141】また、複数の連続したスタートヘッダを検出するとともに、これらのスタートヘッダの間に在る複数のテキストグループに相当するデータ内容を比較して、この比較結果が一致していた場合は、当該テキストグループに相当するテキストデータの有効性を確認することができる。つまり、テキストデータ全体を比較しているので、データの整合性に基づいた有効なテキストデータを得ることができるようになる。
- 40 【0142】さらに、スタートヘッダが所定回数検出された時点で、管理情報の読み込みを終了させて、検出されているスタートヘッダに対応した文字情報の検証処理に移行することで、必要以上のデータの読み込み動作を抑制することができ、システムコントローラにかかる負担を抑制することができるようになる。
- 50 【0143】また、上記したテキストデータの検証処理を行なう場合、管理情報から文字情報とされるテキスト

データのみを抽出して行なうことにより、必要最小限のデータによって検証処理を行なうことができ、処理自体が簡単になるとともに、検証結果が有効であった場合についても、ホストコンピュータに対する転送処理が簡単になる。

【0144】さらに、ディスクからインターリーブされた状態で読み出されたデータ形態の管理情報を、デインターリーブされる前に検出するので、デインターリーブ処理を省略することができるようになる。

【0145】また、前記テキストデータに対する誤り検出符号に基づいて、当該テキストデータに誤りが検出されたときに、スタートヘッダが検出されないまま、それまでに前記テキストデータの誤りが所定回数以上検出されていた場合には、ディスクにテキストデータが記録されていないと判断するようにしている。したがって、必要以上のデータの読み込み動作を抑制したうえで、オーディオCDとCD-TEXTの識別を行うことができるようになる。

【0146】さらに、ディスクにテキストデータが記録されていないと判断した時点で、ディスクに記録されている情報の読み出しを行うバッファリング処理を終了させることで、システムコントローラにかかる負担を抑制することができるようになるという利点がある。

【図面の簡単な説明】

【図1】本発明の実施の形態の再生装置の構成を説明するブロック図である。

【図2】ディスク(CD)のフレーム構造の説明図である。

【図3】ディスク(CD)のサブコーディングの説明図である。

【図4】ディスク(CD)のサブQデータの説明図である。

【図5】ディスク(CD)のTOCデータの説明図である。

【図6】テキストデータの構造を包括的に示す説明図である。

【図7】サブコーディングフレームとテキストデータとの構造的な関係を示す説明図である。

【図8】テキストデータとしてパケットの構造を示す説明図である。

【図9】テキストデータの構造として、シンボル単位のデータからパックを形成する過程を説明するための説明図である。

【図10】パックの構造を示す説明図である。

【図11】ID1の構造を示す説明図である。

【図12】ID1～ID4の構造をそれぞれ示す説明図

である。

【図13】ID1の定義内容を示す説明図である。

【図14】テキストデータとしてトラックの曲名を格納する場合のパックの構造を示す説明図である。

【図15】再生処理時のキャッシュメモリのマッピング例を説明する図である。

【図16】バッファリング処理の行程を説明するフローチャートである。

【図17】図17に続くバッファリング処理の行程を説明するフローチャートである。

【図18】図17のフローチャートに対応して、NGが検出されている場合にサブコードデータのバッファリングを終了させる処理に進む流れを説明するフローチャート。

【図19】NGが検出されている場合にサブコードデータのバッファリングを終了させ、有効性を検出する処理に進む流れを説明するフローチャート。

【図20】再生処理時のキャッシュメモリのローサブコードエリアの内容を模式的に説明する図である。

【図21】キャッシュメモリのローサブコードエリアに格納されているローサブコードデータを説明する図である。

【図22】8ビット／6ビット変換の概要を説明する図である。

【図23】テキストデータ有効性検証処理の行程を説明するフローチャートである。

【図24】図22に続くテキストデータ有効性検証処理の行程を説明するフローチャートである。

【図25】ヘッダタイプの一例を説明する図である。

【図26】2番目のパックのヘッダの検証処理を説明するフローチャートである。

【図27】サイズインフォメーションの検証処理を説明するフローチャートである。

【図28】ローサブコードデータのデータ例を説明する図である。

【図29】テキストデータのデータ例を説明する図である。

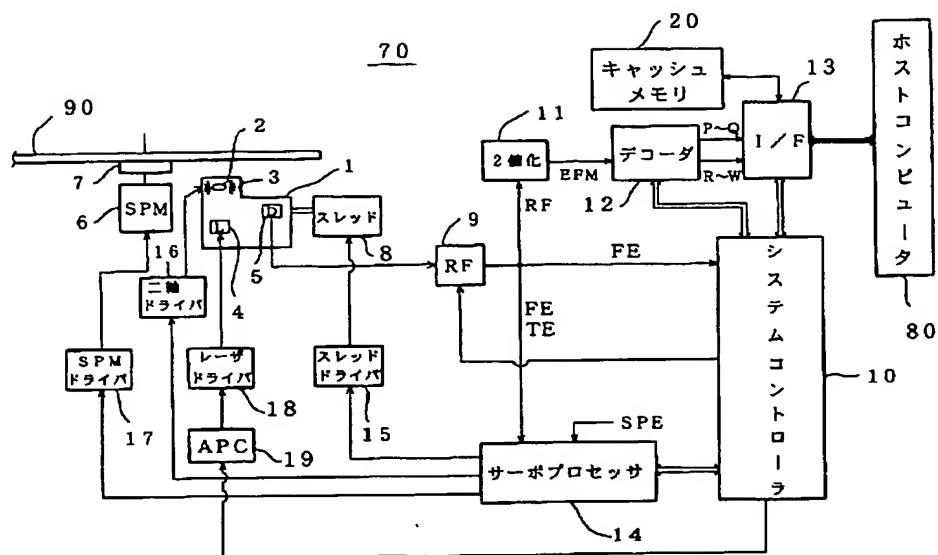
【図30】本発明の変形例を説明する模式図である。

【図31】本発明の他の変形例を説明するフローチャートである。

【符号の説明】

1 ピックアップ、9 RFアンプ、10 システムコントローラ、12 デコーダ、13 インターフェース部、14 サーボプロセッサ、20 キャッシュメモリ、70 再生装置、80 ホストコンピュータ、90 ディスク

【図1】



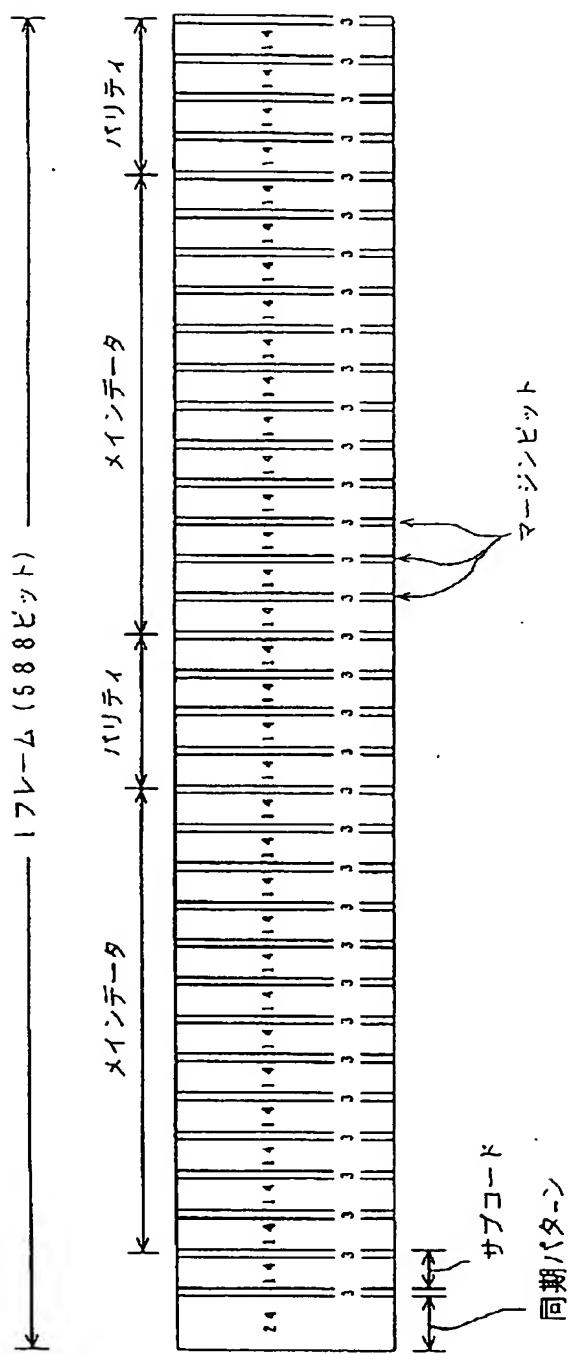
【図3】

フレーム サブコーディングフレーム

(a)	98n+1	同期パターン (S0)							
	98n+2	同期パターン (S1)							
	98n+3	P ₁	Q ₁	R ₁	S ₁	T ₁	U ₁	V ₁	W ₁
	98n+4	P ₂	Q ₂	R ₂	S ₂	T ₂	U ₂	V ₂	W ₂
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
	98n+97	P ₉₅	Q ₉₅	R ₉₅	S ₉₅	T ₉₅	U ₉₅	V ₉₅	W ₉₅
	98n+98	P ₉₆	Q ₉₆	R ₉₆	S ₉₆	T ₉₆	U ₉₆	V ₉₆	W ₉₆
	98(n+1)+1								

(b)	Q ₁ ~ Q ₄	Q ₅ ~ Q ₈	Q ₉ ~	Q ₉₆	Q ₉₇ ~ Q ₉₆
	コントロール	アドレス	サブQデータ		CRC

【図2】



【图4】

[図 5]

リードインエリアでのサブQデータ (TOC)

(a)

トランザクション TNO (00)	POINT	NIN	SEC	PFRAME	0	PMIN	PSBC	PFNAME
	8	8	8	8	8	8	8	8
	72ビット							

トラック #1～#n 及びリードアウトエリアでのサブQデータ

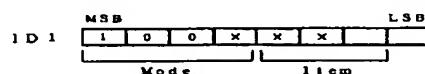
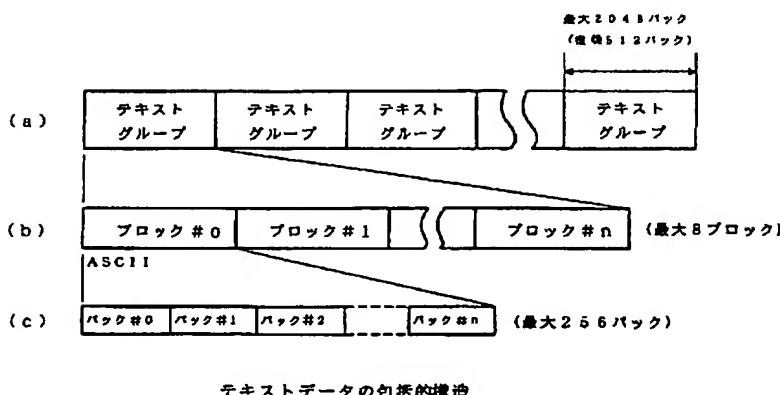
(b)	トラックナン THN (01~99) AA	インプクス X (01~99)	MIN	SEC	FRAME	0	AMIN	ASEC	APNAME
	8	8	8	8	8	8	8	8	8

72ビット

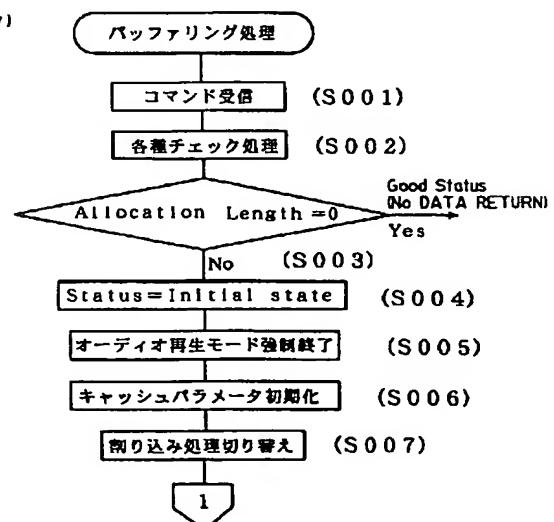
NO	ブロックNo.	POINT	PMIN, PSEC, PFRAME	
00	n	01	00. 02. 32 } トランク#1の	
	n+1	01	00. 02. 32 }	スタートポイント
	n+2	01	00. 02. 32 }	
	n+3	02	10. 15. 12 } トランク#2の	
	n+4	02	10. 15. 12 }	スタートポイント
	n+5	02	10. 15. 12 }	
	n+6	03	16. 28. 63 } トランク#3の	
	n+7	03	16. 28. 63 }	スタートポイント
	n+8	03	16. 28. 63 }	
	n+9	04	* * *	
	n+10	04	* * *	
	n+11	04	* * *	
	n+12	05	* * *	
	n+13	05	* * *	
	n+14	05	* * *	
	n+15	06	49. 10. 03 } トランク#6の	
	n+16	06	49. 10. 03 } スタートポイント	
	n+17	06	49. 10. 03 }	
	n+18	A0	01. 00. 00 } ディスクの最初のトラック	
	n+19	A0	01. 00. 00 } のトラックナンバ	
	n+20	A0	01. 00. 00 }	
	n+21	A1	08. 00. 00 } ディスクの最後のトラック	
	n+22	A1	06. 00. 00 } のトラックナンバ	
	n+23	A1	06. 00. 00 }	
	n+24	A2	52. 48. 41 } リードアウトトラックの	
	n+25	A2	52. 48. 41 }	
	n+26	A2	52. 48. 41 }	スタートポイント
00	n+27	01	00. 02. 32 ↓	くり返す
	n+28	01	00. 02. 32 ↓	
	*	*	* * *	
	*	*	* * *	
	*	*	* * *	

[図 6]

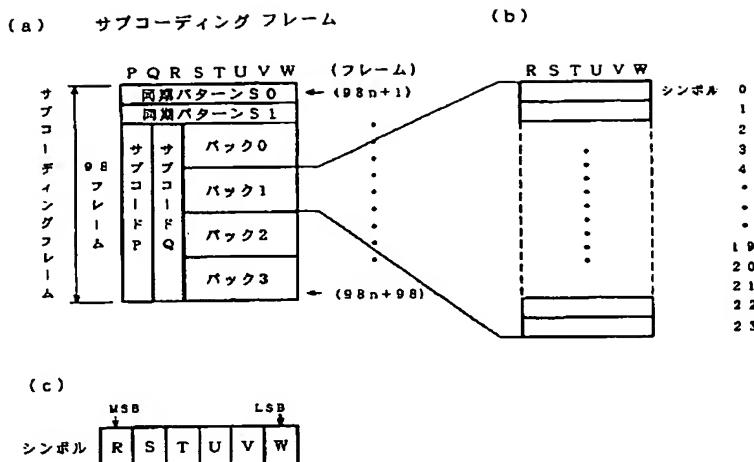
【図11】



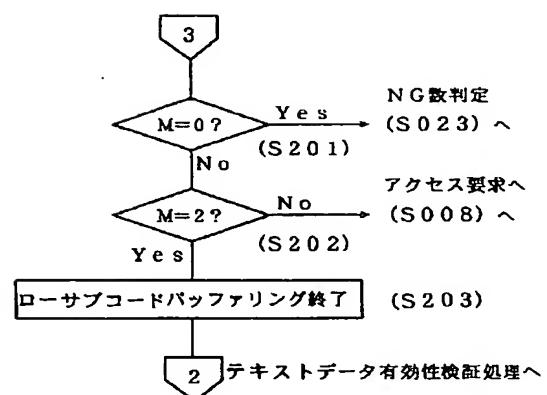
[図161]



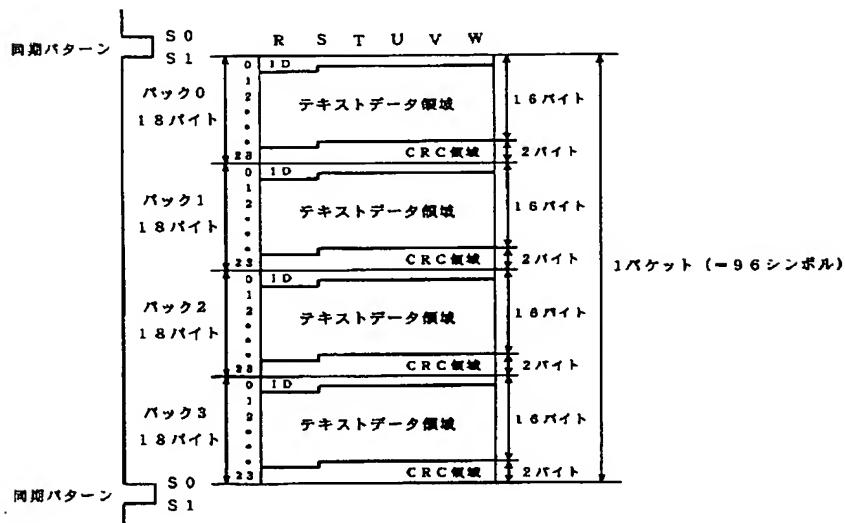
【図7】



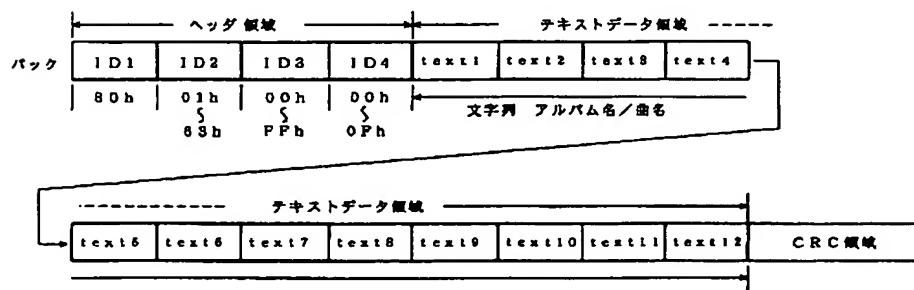
【図19】



【図8】

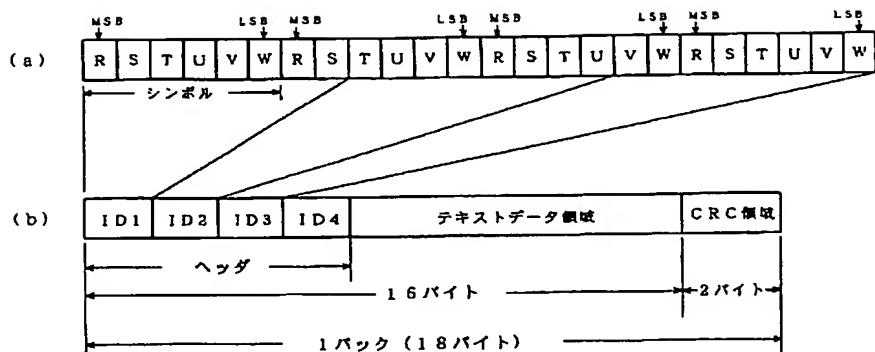


【図14】



トラックの曲名データを格納するパックの構造

【図9】



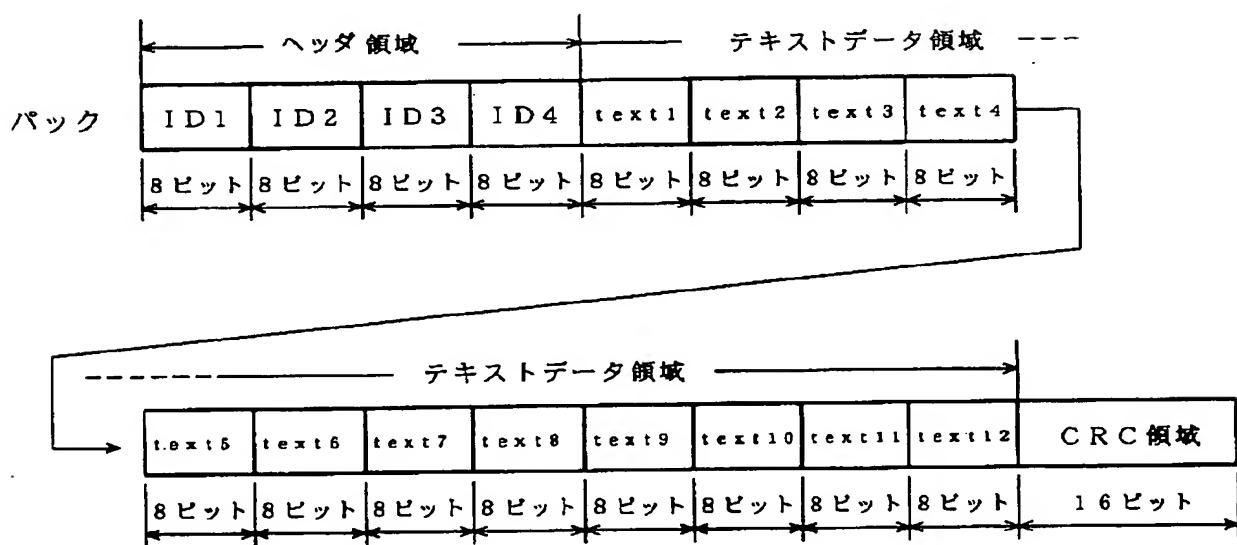
ID1: Pack Type Indicator

ID2: Tracking Number Indicator

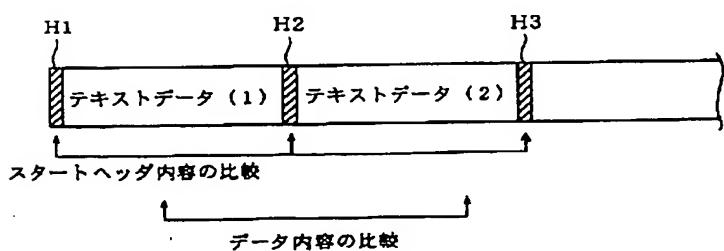
ID3: Sequence Number Indicator

ID4: Block Number and Character Position Indicator

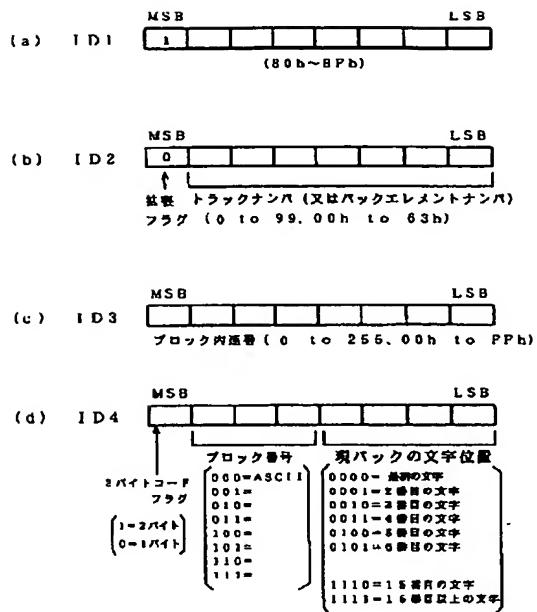
【図10】



【図30】



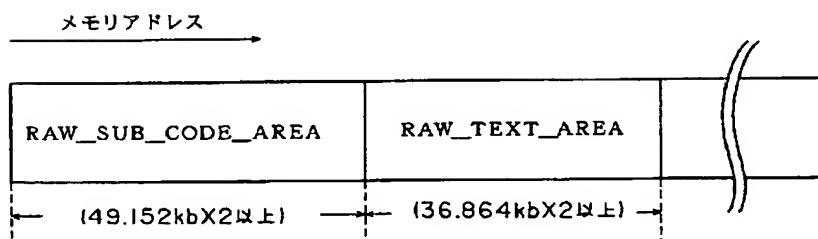
【図12】



【図13】

ID1	識別内容
80h	アルバム名 (ID2=00h) / トラックの曲名 (ID2=01h~63h)
81h	演奏者 / 指揮者 / オーケストラ名
82h	作詞者名
83h	作曲者名
84h	編曲者名
85h	メッセージ
86h	ディスクID
87h	ジャンル
88h	TOC
89h	2nd TOC
8Ah	予約
8Bh	予約
8Ch	予約
8Dh	管理
8Eh	CPC/EAN/ISRC
8Fh	サイズインフォメーション

【図15】

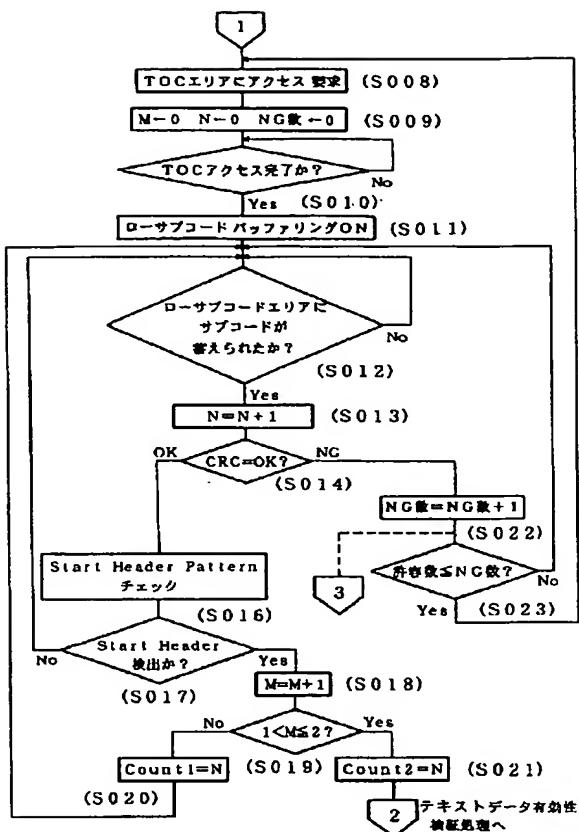


キャッシュメモリマッピング例

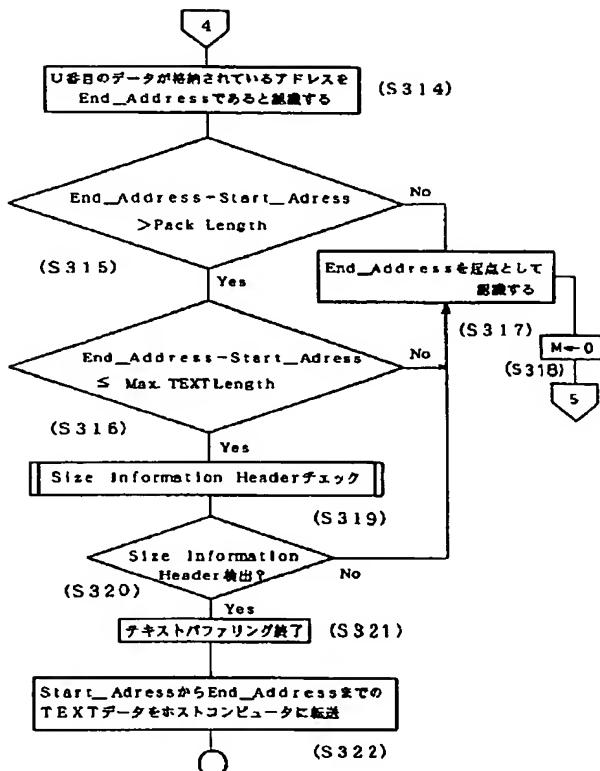
【図21】

Sub-	P	Q	R	S	T	U	V	W	
byte [0]	x	x	1	0	0	0	0	0	→ 10000000b
byte [1]	x	x	0	0	0	0	0	0	→ 00000000b
byte [2]	x	x	0	0	0	0	0	0	→ 00000000b
byte [3]	x	x	0	0	0	0	0	0	
byte [4]	x	x	x	0	0	0	0	0	→ 00000000b / 10000000b
byte [5]	x	x	0	0	x	x	x	x	

【図17】



【図24】



【図20】

(a)

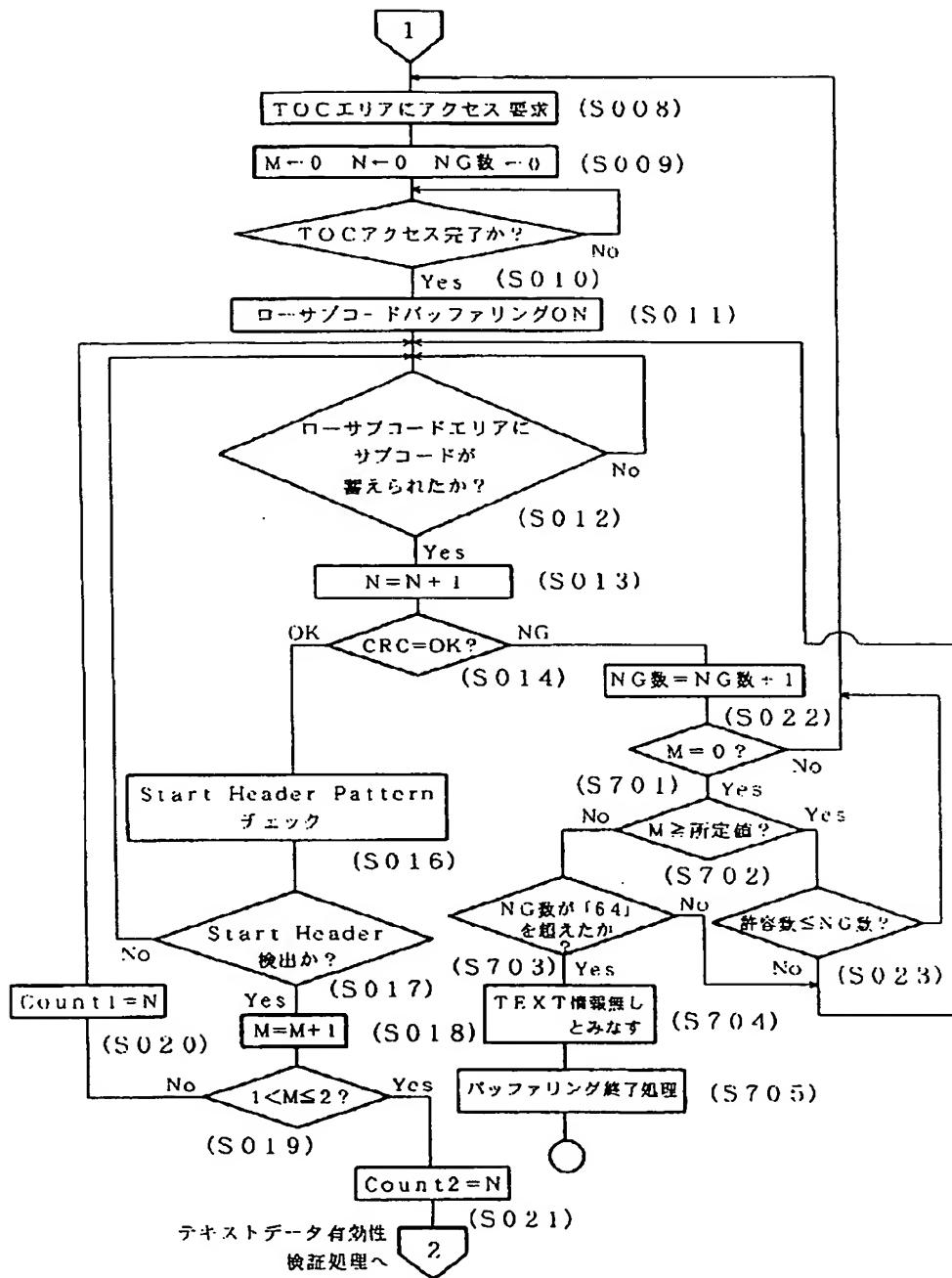
	CRC OK																
N	0	1	2	3	4	5	6	7	8	9	512	513	514	515	516	517
M	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2
CRC NG	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4
Count1	0	0	0	0	0	0	0	7	7	7	7	7	7	7	7	7
Count2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	516	517

(b)

Cache#	1	2	3	4	5	6	7	8	9	512	513	514	515	516	517
CRC NG	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←
Count1=7	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←
CRC OK	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→
Count2=516	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→

テキストグループが存在する領域

【図18】



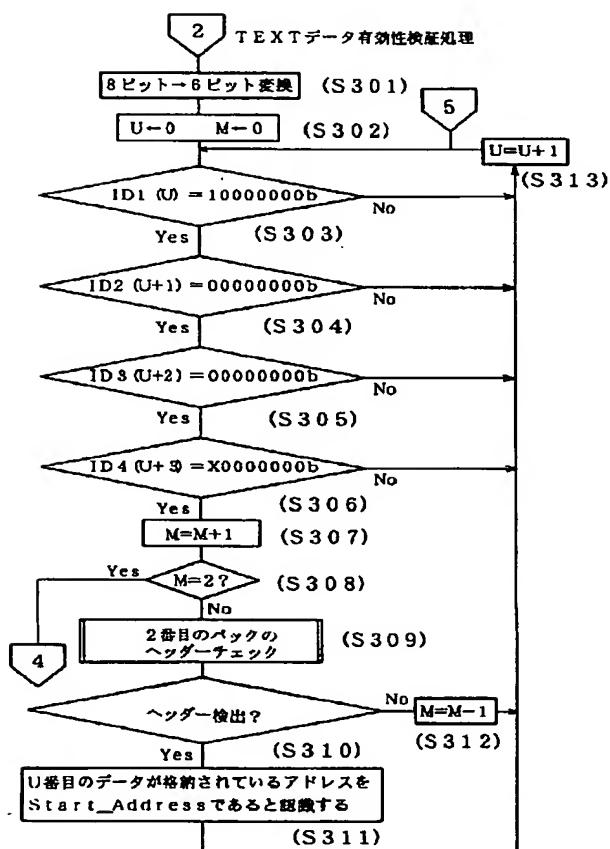
【図22】

Sub-	P	Q	R	S	T	U	V	W		+7	+6	+5	+4	+3	+2	+1	+0
byte [0]	x	x	a	b	c	d	e	f		a	b	c	d	e	f	g	h
byte [1]	x	x	g	h	i	j	k	l		i	j	k	l	m	n	o	p
byte [2]	x	x	m	n	o	p	q	r		q	r	s	t	u	v	w	x
byte [3]	x	x	s	t	u	v	w	x									

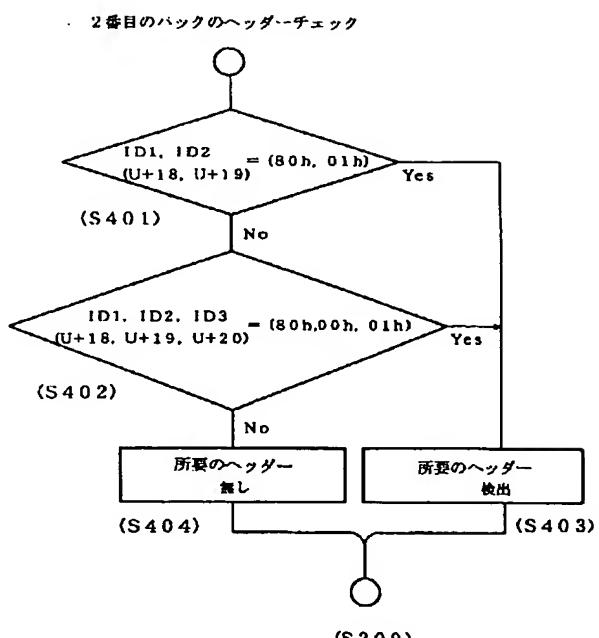
(a)

(b)

【図23】



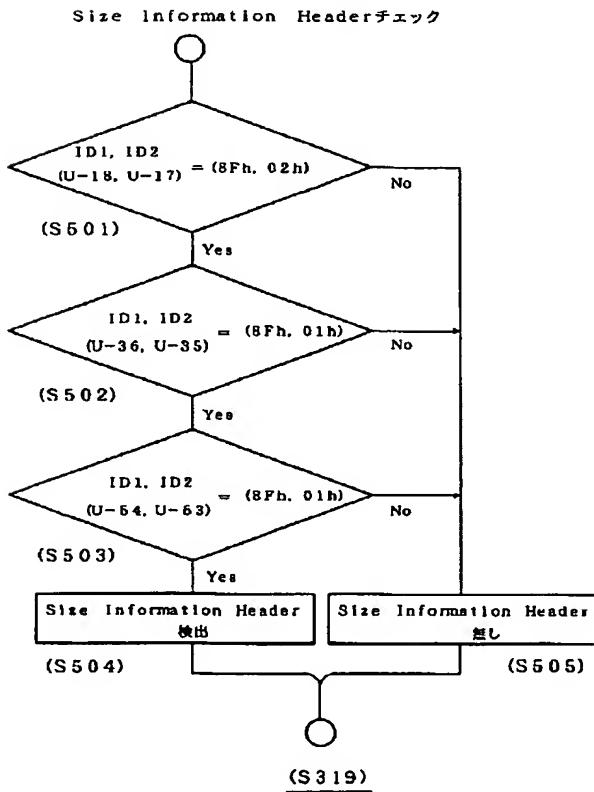
【図26】



【图 25】

	ID1	ID2	ID3	ID4	
(a)	80h	00h	XXh	XXh	Title of Album name
	80h	01h	XXh	XXh	Track Title #01h
(b)	80h	02h	XXh	XXh	Track Title #02h
	80h	:	:	:	Track Title #h
	80h	63h	XXh	XXh	Track Title #63h
	81h	XXh	XXh	XXh	Names of the performers
	:				
(c)	8Fh	00h	XXh	XXh	Size Information #1
	8Fh	01h	XXh	XXh	Size Information #2
	8Fh	02h	XXh	XXh	Size Information #3

【図27】



【図28】

Hex deci +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F 0123456789ABCDEF
00000h 000000 20 00 18 09 00 05 00 54 10 15 11 05 08 04 3D 06T.....=.
00010h 000016 08 05 00 08 53 33 36 20 20 00 18 04 03 04 4D 08S36M.
00020h 000032 00 05 01 41 14 65 51 59 08 44 45 0C 13 45 53 14 ...A..oY..DE..ES.
00030h 000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00360h 000864 [20 00 00 00] 00 03 20 70 09 37 0C 20 10 14 31 09p.7. ...1.
00370h 000880 15 24 14 60 0C 29 2F 73 20 00 00 01 03 02 40 2D ..\$..)../s8.
00380h 000896 08 05 09 45 51 02 00 20,05 11 05 13 00 27 32 ..E0...*2
00390h 000912 20 00 04 02 00 34 39 20 12 44 15 12 08 04 05 4240.D.....8
003A0h 000928 13 35 55 14 08 4A 60 2D 20 00 04 03 03 34 25 14 ..5U..J.....4K.
003B0h 000944 00 04 11 11 41 53 24 5D 45 14 22 41 1A 13 7D 20 44 ..AS\$!E..A..!J..D
003C0h 000960 20 00 08 04 02 14 38 45 00 04 21 0F 13 04 11 09BE..!.....
003D0h 000976 13 24 1C 60 13 3E 07 66 20 00 0C 05 02 15 55 14 ..\$..>..IU.
003E0h 000992 08 04 19 4F 54 22 01 01 08 04 21 05 14 26 12 4F ..07%....!..8.0
003F0h 001008 20 00 0C 06 03 34 3C 00 12 44 05 1A 16 12 01 53 ...4<.D.....S
00400h 001024 12 04 45 04 11 5D 46 16 20 00 10 07 02 22 01 0F ..E..]F.
00410h 001040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00420h 001056 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00430h 001072 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00440h 001088 00 00 00 40 00 47 1E 2F 23 30 44 28 00 40 00 40 ..0.E../0D+..9.0
00450h 0010A4 40 00 00 00 00 40 40 03 08 00 00 00 00 04 77 01 E.W.
00460h 0010B0 23 30 08 2C 00 00 00 40 00 00 00 09 00 00 00 00 00 000..
00470h 0010C6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00480h 0010E2 09 37 0C 60 50 14 31 09 15 24 14 60 0C 29 2F 73 ..\$..*1..)../s
00490h 0010F8 20 00 00 01 03 02 00 20 08 45 09 45 11 42 40 2D-E..E..B0.
004A0h 0010H4 00 05 11 45 13 04 27 32 20 00 44 02 00 74 30 60 ...E..H'2..D..10*
004B0h 002000 12 04 55 12 48 44 05 02 13 75 15 54 08 0A 60 6D ..U..ID..U..T. 'a

RAW SUB CODE DATA

[図29]

CD-TEXT DATA

【図31】

